



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERÍAS INDUSTRIALES

Grado en Ingeniería Electrónica Industrial y Automática

**Control de velocidad de motores BLDC,
mediante sistemas basados en Arduino
para aplicaciones en vehículos eléctricos**

Autor:

Rodríguez Pintor, Jesús Carlos

Tutor:

Herrero de Lucas, Luis Carlos

Tecnología Electrónica

Valladolid, febrero 2015.





Dedicado a mí familia por su apoyo incondicional desde el inicio, quienes me han inculcado que todo lo que merece la pena requiere un sacrificio.

Gracias a todos los que me han prestado su ayuda y se han interesado en las consecuencias de un futuro con vehículos eléctricos.

Pero quiero hacer una mención especial a la persona que me ha tutelado y supervisado la evolución del TFG con dedicación y paciencia; quién no vaciló ni un instante en dirigirme en esta ardua tarea cuando se lo pedí.



[PÁGINA INTENCIONADAMENTE EN BLANCO]



“Los problemas más importantes y urgentes de la tecnología actual ya no son las satisfacciones de las necesidades primarias o de los deseos arquetípicos, sino la reparación de los males, daños y perjuicios provocados por la tecnología de ayer”.

Dennis Gabor

(Físico y Premio Nobel)



[PÁGINA INTENCIONADAMENTE EN BLANCO]



RESUMEN

Ante el actual auge e inminente crecimiento del sector del vehículo eléctrico (VE), como consecuencia de una clara serie de ventajas que presenta, es necesario centrarse e innovar en este campo del conocimiento; puesto que según las previsiones, en unos años el protagonismo que van a alcanzar este tipo de automóviles será prácticamente absoluto. Haremos un breve recorrido a través de la historia del VE, desde sus comienzos; pasando por su situación actual, y por último qué cabe esperar en el futuro sobre este tipo de tecnología. Nos centraremos principalmente en la manera de cómo controlar un motor de tipo BLDC con un microcontrolador Arduino, mediante la técnica de modulación de pulsos o PWM. En un primer momento se simulará el comportamiento del sistema con la herramienta Simulink del programa Matlab. Posteriormente, crearemos un circuito físico con el que controlaremos un motor BLDC real.

Facing the current boom and imminent growth of the electric vehicle sector (VE), due to its advantages, we have to focus and innovate in this field of knowledge; because according to predictions, the leadership that this kind of cars is going to achieve in a few years is almost complete. We will make a brief tour through the history of VE, since its inception; through its current situation, and finally we will deal with the future of this type of technology. We will mainly focus on how to control a BLDC motor with an Arduino microcontroller, using the pulse modulation PWM technique. At first, the performance of the system will be simulated with the Matlab Simulink tool. Finally, we will create a physical circuit that we will use to control a real BLDC motor.

-Las palabras clave para definir este trabajo fin de grado son:

Control PWM, Motor BLDC, Arduino, Vehículo Eléctrico, Control de velocidad

PWM controlling, BLDC motor, Arduino, Electric Vehicle, Speed Control



[PÁGINA INTENCIONADAMENTE EN BLANCO]



ÍNDICE

Resumen	7
1.-Introducción	15
1.1 Breve reseña histórica del VE	17
1.2 Ventajas e inconvenientes del VE	19
1.3 Perspectivas de futuro del VE	21
1.4 Objetivos	22
2.- Descripción de los componentes del sistema electrónico	25
2.1 Sistema de control mediante Arduino	25
2.2 Convertidor DC/AC modelo L6234	38
2.3 Motor BLDC	44
2.4 Sensores tipo Hall	55
3.- Desarrollo del controlador de velocidad para motor BLDC	59
4.- Simulación con Matlab	89
5.-Conclusiones	99
6.-Fuentes	101
Bibliografía	101
Páginas web/enlaces	101
Anexos	103
Datasheet Arduino	104
Datasheet Inversor	115
Datasheet Motor	131
Apéndice: Glosario de abreviaturas y acrónimos	135



Índice de figuras

Fig.1.1-Polución debida al tránsito de vehículos	16
Fig. 1.2- Thomas Alva Edison con un VE (1890)	17
Fig.1.3- “La Jamais Contente”	18
Fig. 1.4-Gráfica nº VE y VH (en millones) al año	22
Fig. 1.5- Esquema de un sistema electrónico genérico	22
Fig. 2.1- Localización de Arduino en el sistema electrónico	25
Fig. 2.2. Placa Arduino	26
Fig. 2.3-Pines de alimentación de Arduino	27
Fig. 2.4-Pines digitales de Arduino	28
Fig. 2.5- Pines analógicos de Arduino	29
Fig. 2.6. Puertos de la placa Arduino	29
Fig.2.7- Pantalla de inicio software Arduino	30
Fig. 2.8-Herramientas del software	31
Fig. 2.9- Cable USB para conexión placa Arduino-PC	31
Fig. 2.10-Variedad de duración de ancho de pulso	34
Fig. 2.11-Posición del convertidor DC/AC en el sistema Electrónico	38
Fig. 2.12- Esquema de un inversor genérico	39
Fig. 2.13-Inversor integrado L6234	39
Fig. 2.14- Función de las patillas del L6234	40
Fig. 2.15-Esquema interno del L6234	40
Fig. 2.16- puertas And del L6234	41
Fig. 2.17- drivers del L6234	42
Fig. 2.18- transistores DMOS del L6234	42
Fig. 2.19- Rama 2, de las tres que posee el inversor	43



Fig. 2.20- Puesto que ocupa el motor en el sistema electrónico	44
Fig. 2.21- Tipo de máquina eléctrica en función del sentido de la corriente	45
Fig. 2.22- Esquema tipos de motores	45
Fig. 2.23-Ejemplos de motores DC	47
Fig. 2.24-Armadura de un motor DC	48
Fig. 2.25-Conmutador de un motor DC	49
Fig. 2.26-Campo de un motor DC	49
Fig. 2.27-Rotor (izq) y estator (der) de un motor BLDC	51
Fig. 2.28-Configuración BLDC, 3 bobinas (a), 6 polos de potencia (b)	51
Fig. 2.29-Conjunto de fuerzas electromagnéticas que intervienen en la Fuerza de Lorentz	52
Fig. 2.30- Secuencia de giro del motor BLDC	54
Fig. 2.31-Puesto de los sensores en el sistema electrónico	55
Fig. 2.32-Representación gráfica del efecto Hall	56
Fig. 2.33-Componentes de un sensor Hall	57
Fig. 2.34 Sensor con transistor en colector abierto	57
Fig. 2.35-Ejemplo de Sensor Hall	58
Fig. 3.1-Conjunto que conforma el sistema electrónico	59
Fig. 3.2- Esquema de montaje del controlador	61
Fig. 3.3-Vista superior del controlador	62
Fig. 3.4-Vista inferior del controlador	62
Fig. 3.5-Conexión entre el inversor y las bobinas del motor	63
Fig. 3.6-Corrientes al aplicar vector 101	64
Fig. 3.7-Nomenclatura de las ramas y transistores del inversor	65
Fig. 3.8-Nomenclatura del inversor real	66



Fig. 3.9-Recordatorio del concepto PWM	67
Fig. 3.10-Forma de onda de las fases A-B del motor obtenidas con el osciloscopio	69
Fig. 3.11-Forma de onda de las fases B-C del motor obtenidas con el osciloscopio	69
Fig. 3.12-Diagrama de bloques del programa principal	70
Fig. 3.13-Vista conjunto que conforma el controlador del motor, el motor y la alimentación	76
Fig. 3.14-nivel de tensión a la entrada de la placa	76
Fig. 3.15-forma de onda de dos sensores a bajas revoluciones del motor	77
Fig. 3.16-forma de onda de los sensores a altas revoluciones del motor	77
Fig. 3.17-forma de onda del PWM en EN1	78
Fig. 3.18-forma de onda de los impulsos aplicados a IN1 e IN2	78
Fig. 3.19- Sentido de circulación de la corriente en el frenado regenerativo	79
Fig. 3.20- Diagrama de bloques programa con frenado regenerativo	80
Fig. 4.1- Representación batería Simulink	90
Fig. 4.2- Representación inversor simulink	90
Fig. 4.3- Representación motor simulink	91
Fig. 4.4- Representación bloque decoder simulink	91
Fig. 4.5- Configuración interna bloque decoder	92
Fig. 4.6- Representación bloque gates simulink	92
Fig. 4.7- Configuración interna del bloque gates	93
Fig. 4.8- Bucle realimentación PI	94
Fig. 4.9- Representación bloque PWM simulink	94
Fig. 4.10- Configuración interna bloque PWM	95
Fig. 4.11- Modelo simulador simulink	96
Fig. 4.12- Vcontrol, pulsos, Gate y Salida	97



Índice de tablas

Tabla. 2.1- Tabla de la verdad del inversor L6234	43
Tabla 2.2- Comparativa de motores DC con y sin escobillas	55
Tabla 3.1- Relación vectores-activación fases (bobinas)	64
Tabla 4.1-Tabla de vectores con su correspondiente activación de bobinas	91
Tabla 4.2-Correspondencia entre activación de bobinas y polos de potencia	93

Índice de ecuaciones

Ec. 2.1- Fuerza de Lorentz	59
----------------------------	----



[PÁGINA INTENCIONADAMENTE EN BLANCO]



1

INTRODUCCIÓN

La sociedad contemporánea a la que pertenecemos viene caracterizada por una creciente demanda de bienestar.

A lo largo del siglo XX ha ido aumentando progresivamente la necesidad de un consumo masivo de recursos. Como consecuencia de lo anterior, el consumismo generado ha llegado a comprometer seriamente los recursos naturales y una economía sostenible.

Dentro de esta dinámica, la principal fuente generadora de la energía necesaria para satisfacer las demandas de esta sociedad se ha basado en el petróleo y el gas natural.

De todos son conocidas las consecuencias del consumo descontrolado de estos recursos: una creciente y nociva contaminación medioambiental, una cada vez mayor y más peligrosa dependencia de los mismos de numerosos países que no los poseen, lo que llega a afectar de forma evidente a sus economías, llegando a ser, incluso, la causa que ha desencadenado numerosos conflictos internacionales.

Como reacción a la situación generada, en la actualidad va en aumento una concienciación sobre la necesidad de una reducción de consumo energético y en la investigación y búsqueda de fuentes alternativas de energía.

El vehículo automóvil que, sin lugar a dudas constituye un elemento clave en el funcionamiento de la sociedad moderna, sobre el cual se basa en gran parte el sistema de transporte de personas y recursos, ha ido evolucionando notablemente, perfeccionándose su rendimiento y mejorándose permanentemente sus prestaciones, pero aún sobre la base de un numeroso y activo conjunto de vehículos propulsados por la combustión de hidrocarburos (ciclomotores, automóviles y camiones), lo que hace que persistan los mismos problemas en cuanto al elevado consumo de energía procedente de este recurso a la vez que supone una de las más importantes causas de la contaminación ambiental¹.

De hecho, en Europa, la mayoría de los países no disponen de reservas de petróleo ni de gas natural; sin embargo, en la actualidad casi el 100% del transporte terrestre se efectúa usando el petróleo como combustible.

¹ (Véase fig.1.1)La contaminación generada por los automóviles se concentra sobre todo en el aire.



Al igual que para el resto de ingenios que participan en el bienestar de la sociedad y que consumen algún tipo de energía, se ha incrementado la búsqueda de alternativas para sustituir los motores de combustión, manteniendo e, incluso, aumentando sus capacidades y rendimiento, como puede ser emplear carburantes sustitutivos de la gasolina, desarrollar otras fuentes energéticas alternativas tales como la eléctrica, etc.



Fig.1.1-Polución debida al tránsito de vehículos

El objetivo es desarrollar un sustituto al actual vehículo que contribuya al bienestar económico y social sin agotar los recursos naturales, ni dañar la salud humana, preservando, a la vez, el medio ambiente.

El *vehículo eléctrico* (junto con el híbrido [VH]) constituye una de las alternativas para la resolución del problema expuesto anteriormente, siendo la única tecnología disponible que no genera ningún tipo de emisión y, por lo tanto, no contamina en absoluto donde circula.

Sin embargo, en la actualidad, la *tecnología del vehículo de motor eléctrico* está aún en proceso de investigación que tiene como reto superar los numerosos problemas que esta plantea, como puede ser el logro de una fuente de energía más eficaz que facilite unas prestaciones similares a las que proporcionan los vehículos de combustión, el alto desembolso inicial, la inexistencia de infraestructuras, una producción en serie comparable en cifras, rendimiento, vida útil, etc.

Con el fin de determinar los objetivos a lograr en el proceso de investigación relativo al perfeccionamiento del vehículo de motor eléctrico, resultará importante conocer la evolución de dicho proceso a lo largo de la historia para ver los obstáculos que han ido apareciendo y cómo se han subsanado (si es el caso), las ventajas e inconvenientes que presenta en la actualidad y las perspectivas existentes hacia un futuro próximo.

1.1. BREVE RESEÑA HISTÓRICA DEL VE

La historia del vehículo eléctrico se remonta a finales del siglo XIX, cuando otras tecnologías de vehículos a motor comenzaban vagamente a hacer su aparición.

Los avances logrados en el motor de combustión acabaron desbancando finalmente al motor eléctrico hasta que la sociedad ha tomado conciencia sobre la necesidad de encontrar otra alternativa debido principalmente a los problemas medioambientales y energéticos a los que está sometida.

El origen del vehículo eléctrico no acaba de estar muy claro. Dependiendo de la fuente documental a la que se consulte, se puede atribuir a un hombre de negocios escocés (Robert Ander) entre los años 1932 y 1939, o bien, al profesor Strating en 1935, en la ciudad de Groninberg (Alemania).

En cualquiera de los casos, se puede decir que su verdadero impulso tuvo su origen en Escocia, allá por 1893, debiendo competir ya con los vehículos a vapor que se fabricaban en Inglaterra por aquél entonces.



Fig. 1.2- Thomas Alva Edison con un VE (1890)

En los años setenta de ese mismo siglo, se perfeccionó el motor, siendo este más ligero y evolucionado. La velocidad punta que alcanzaba era de 13 km/hora, muy inferior a la alcanzada por los vehículos competidores de vapor, así como su autonomía. El motivo de estas desventajas hay que atribuirlo al elevado peso de las baterías que empleaba como fuente de energía.

Posteriormente se fueron superando estos inconvenientes, logrando unas sensibles mejoras en las prestaciones, de forma que ya en 1899 se produjeron dos mil vehículos eléctricos en Estados Unidos para ser empleados como taxis en Nueva York, Chicago y Boston. En este año el coche eléctrico llamado “La Jamais Contente” (ver fig. 1.3) alcanzó los 100 km/h, un record mundial.

Un año más tarde, en 1900, ya se producían cuatro mil vehículos, compitiendo con los de vapor y gasolina. Paradójicamente, los de gasolina no eran bien aceptados debido al ruido que desataban y las vibraciones que producían.



Fig.1.3- “La Jamais Contente”

En esta carrera la mayor profusión del vehículo eléctrico se alcanzó en 1912, año en que los avances logrados permitieron unas mejoras considerables en su rendimiento y confort, como las cubiertas de goma maciza y una velocidad punta de 60 km/hora.

Pero en 1920, las prestaciones que ofrecían los vehículos de combustión interna superaban considerablemente a las aportadas por el motor eléctrico. El de gasolina es el que se considera con mayores posibilidades de **desarrollo a gran escala y reducción de costes**.

De esta manera, el proceso de mejora del vehículo eléctrico queda prácticamente estancado hasta 1970, año en que debido a una crisis de petróleo, los países incrementan el interés por retomar esta tecnología y comienzan a desarrollar diversos programas para potenciar el vehículo eléctrico como alternativa de los clásicos de combustión interna.



En 1976 ya se presenta en estados Unidos una ley para la investigación, desarrollo y demostración del vehículo eléctrico e híbrido. Desde entonces se aceleran las iniciativas, tanto de organismos oficiales como del sector privado para hacer que el vehículo eléctrico sea una realidad.

La **evolución tecnológica de los componentes del vehículo eléctrico** ha sido en los últimos años espectacular. El avance en la tecnología de semiconductores ha permitido la aplicación de los motores de inducción (que elimina el problema de desgaste de los colectores y escobillas de los primeros motores de corriente continua).

Otro aspecto importante en el vehículo eléctrico es el del **control de la velocidad**, donde el primitivo sistema, que variaba la corriente por conexión y desconexión de resistencias en serie, fue sustituido por los rectificadores de silicio y los tiristores.

A fines de la década de 1990 y a principios de 2000, parecía que el automóvil eléctrico se convertiría en una realidad. Pero las limitaciones de la tecnología de baterías fueron excesivas.

Por ahora, los vehículos eléctricos han demostrado que pueden llegar a autonomías medias del orden a los 160 km. La aparición de nuevas tecnologías de baterías como la de níquel metal hidruro y, sobre todo las de litio, hace que ciertas marcas de coches apuesten por este tipo de vehículo junto con el híbrido, si bien este como una alternativa provisional.

La industria automovilística mantiene su compromiso con los vehículos eléctricos. Los expertos creen que los avances en las baterías Li-ion resolverán muchos de estos problemas y que los consumidores tendrán una idea diferente de los vehículos eléctricos.

1.2. VENTAJAS E INCONVENIENTES DEL VE

Los vehículos eléctricos presentan una serie de ventajas ante los actuales vehículos de combustión, pero se debe hacer frente a una serie de inconvenientes si se pretende que estos vehículos acaben siendo realmente la alternativa a los segundos.

Se ha considerado importante enumerar estas ventajas e inconvenientes, de forma que sirva como referencia para defender estos ingenios tecnológicos en el primer caso y, en el segundo, para orientar los diversos procesos de investigación con el fin de perfeccionar progresivamente sus prestaciones y hacerlos competitivos en el mercado.



a). Ventajas

- La utilización de una energía alternativa permite prescindir de combustible, lo que supone ahorrar en una fuente de energía limitada, y/o emplearla para otras necesidades también necesarias, evitando al mismo tiempo la dependencia de la misma.
- El motor eléctrico presenta una mayor eficiencia que el de los de combustión interna.
- Se reduce considerablemente la contaminación atmosférica, especialmente en los grandes entornos rurales.
- Se reduce la contaminación acústica.
- Los costes de mantenimiento son menores.
- Pueden suprimirse algunos sistemas complementarios, como el embrague y la caja de cambios.
- Tienen mayor fiabilidad que los vehículos de combustión ya que, tanto el motor como los sistemas complementarios son más sencillos, lo que, además, facilita la detección de averías, reparación y puesta a punto, desde la perspectiva de mantenimiento mecánico.
- En general requiere menor mantenimiento.
- Su conducción es más fácil.
- Presentan una mejor regulación de par y velocidad al estar controlados por medios eléctricos.
- Disponen de una mayor capacidad de respuesta, en especial en lo que se refiere a la aceleración.
- Tiene capacidad para recuperar parte de la energía durante las aceleraciones.
- El dispositivo de frenado tiene mayor capacidad regenerativa de la potencia absorbida.

b). Inconvenientes

- La inversión inicial para la organización de las cadenas de producción resultará muy alto, lo que provocará que el coste de compra inicial, a su vez resulte muy elevado.
- Hasta la fecha, su autonomía es muy reducida y limitada a la capacidad de las baterías. El elevado peso de estas reduce considerablemente su rendimiento.
- La velocidad punta es sensiblemente menor que el de los vehículos de combustión interna.



- Se requieren instalaciones adaptadas para la recarga de las fuentes de alimentación, por lo que se precisa una infraestructura de suministro de electricidad totalmente nueva en las carreteras.
- Se requiere la actualización del personal especialista (mecánicos) en este tipo de vehículos.

Además, se deberá tener en cuenta que algunos vehículos eléctricos seguirán contaminando ya que, en ciertos casos, la electricidad utilizada para recargar las baterías requerirá el consumo de combustibles fósiles, por lo que hará necesario recurrir a otras fuentes de energía más limpias.

1.3. PERSPECTIVAS DE FUTURO DEL VE

Pese a que, en estos momentos, las marcas tienden a la hibridación de los vehículos, se prevé que acaben siendo totalmente eléctricos. Se debe considerar, por lo tanto, que los VH son una transición entre los de combustión y los eléctricos ya que tienen una configuración compleja, necesitan mucho mantenimiento y tienen un alto coste.

Actualmente existen muchos VE en desarrollo en todo el mundo. Las principales marcas japonesas han anunciado que tendrán automóviles eléctricos en el mercado en un futuro cercano.

China pretende activar la economía del VE, disponiendo en su flota de unos 500.000 en el año 2014 y proyectados 5.000.000 en el 2020. Pero estas previsiones no se están materializando ya que en el 2014 solamente se vendieron 8.000, lo cual dista mucho del objetivo marcado.

Están empezando a surgir nuevas ideas, como la que ha tenido la empresa israelí “Better Place”, que consiste en el intercambio de baterías en un punto en lugar de recargarlas. Se ha aliado con “Chery”² para fomentar su uso y paliar el efecto de la baja autonomía de esta fuente de energía.

En España se pretende alcanzar la cifra de 15.000 en el año 2015. También pretenden desarrollar aproximadamente 300 puntos de recarga en la vía pública para el próximo año. Además de estos, se intentará que haya otros 3000 particulares (en domicilio) y 600 más en parkings.

En la imagen se presenta un gráfico ilustrativo³.

² “Chery” es una firma de automóviles china.

³ Gráfico extraído de “The Economist”(ver fig 1.4).

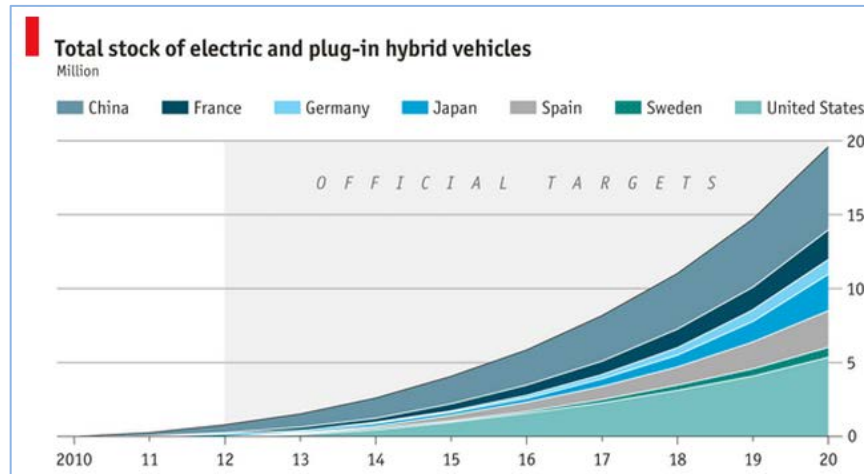


Fig. 1.4-Gráfica nº VE y VH (en millones) al año

Se puede apreciar las previsiones de VE-VH en diferentes países a lo largo de los años como crece de manera exponencial.

1.4. OBJETIVOS

Con este proyecto se pretende diseñar un **sistema electrónico**⁴ real que permita controlar la velocidad de un motor eléctrico de tipo BLDC mediante la técnica PWM. Dicho control estará basado en un controlador “Arduino”. Realizaremos una simulación del un modelo también basado en PWM y verificaremos que el comportamiento corresponde con el real

Con el fin de facilitar el seguimiento del documento, aunque en posteriores capítulos se tratará cada bloque con más detalle, seguidamente se describe brevemente de cada uno de ellos:

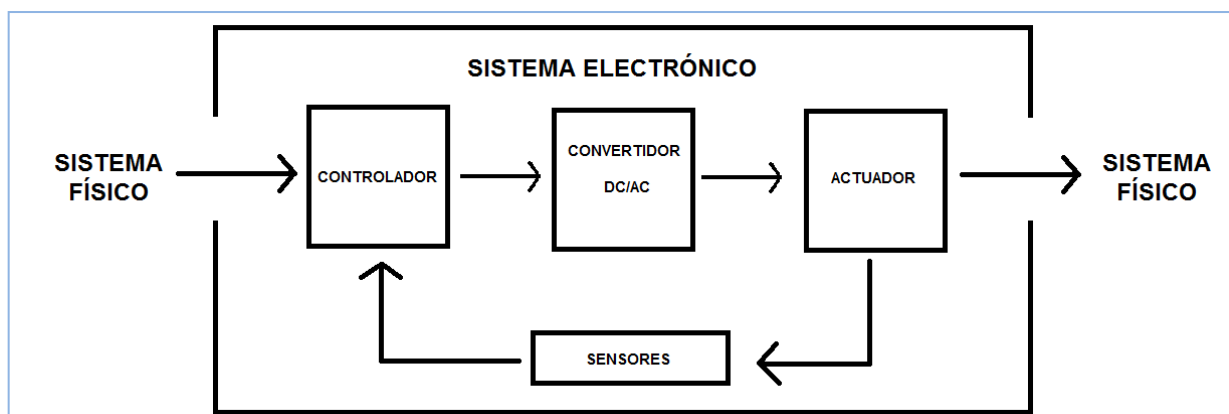


Fig. 1.5- Esquema de un sistema electrónico genérico

⁴ Se entiende por *sistema electrónico* el conjunto formado por controlador, convertidor DC/AC, actuador y sensores.



- ***Controlador***

Manipula una señal eléctrica en sus circuitos internos y según el código que tenga introducido previamente al arranque del sistema; enviará al siguiente bloque una determinada señal codificada. Diseñaremos un programa encargado de ordenar al bloque que le sigue como generar pulsos de señal modulada e indagaremos en el funcionamiento del microcontrolador Arduino.

- ***Convertidor DC/AC*** (o inversor)

Será el encargado de codificar físicamente las tensiones convirtiéndolas en las pulsantes necesarias para que gire convenientemente el motor.

- ***Actuador***

Será el motor. En este apartado haremos primeramente un estudio genérico de los motores eléctricos y, seguidamente, nos centraremos en los motores de corriente continua sin escobillas BLDC.

- ***Sensores***

Adquieren información del mundo físico y la convierten en una señal eléctrica, de manera que el controlador pueda procesarla. Definiremos en qué consiste el *efecto Hall* y sus aplicaciones.



[PÁGINA INTENCIONADAMENTE EN BLANCO]

2

DESCRIPCIÓN DE LOS COMPONENTES DEL SISTEMA ELECTRÓNICO

2.1. SISTEMA DE CONTROL MEDIANTE ARDUINO

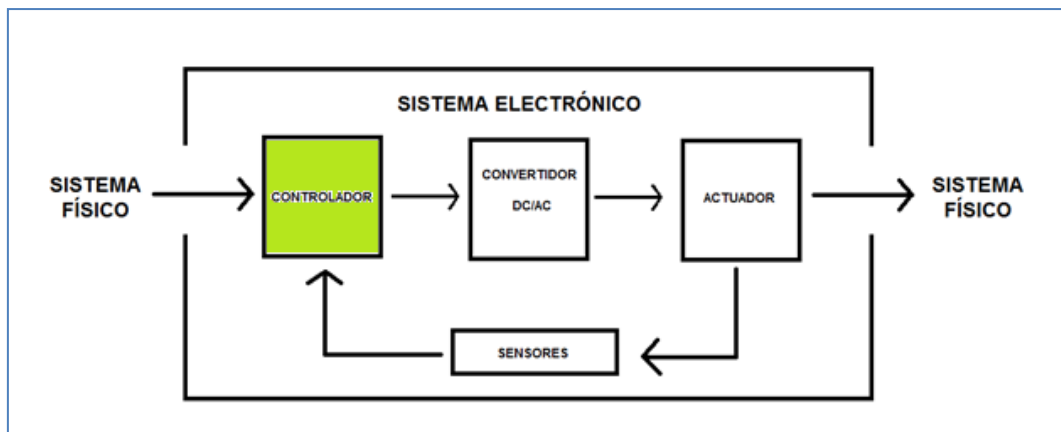


Fig. 2.1- Localización de Arduino en el sistema electrónico

Arduino es, básicamente, una plataforma electrónica que constituye un dispositivo o herramienta que permite una conexión entre el mundo virtual (digital) y el real (físico).

Es una creación relativamente reciente, dentro del cambiante mundo de la electrónica. Su origen se remonta al año 2005 cuando el Instituto de Diseño Interactivo de Ivrea (Italia), realizaba estudios sobre la interacción entre las personas y otros dispositivos. Al instituto le surgió la necesidad de utilizar uno de esos dispositivos, pero de bajo coste económico que permitiera disponer de varios en sus aulas y que desde cualquier ordenador, independientemente de su sistema operativo, se utilizara sin problemas de incompatibilidad. Los primeros se elaboraron para uso interno del instituto, pero este se clausuró ese mismo año.

Para evitar que este proyecto cayera en el olvido, se liberó su patente de forma que todo el mundo tuviera la oportunidad de abordar los problemas que planteara, mejorarlo y crear nuevas versiones.

El equipo que creó este proyecto se denominó “Arduino team” y estaba compuesto principalmente por:

- Massimo Banzi, profesor en ese momento del instituto Ivrea.
- David Cuartielles, zaragozano, profesor de la Escuela de Artes y comunicación de la universidad de Malmö (Suecia).
- David Mellis, en aquella época estudiante en el instituto Ivrea y actual investigador en High-Low Teach.
- Gianluca Martino, responsable de la empresa fabricante de los prototipos de las placas.
- Tom Igoe, profesor de la escuela Arte Tisch.

El *Arduino* está compuesto por tres partes:

- *Hardware.*
- *Software.*
- *Lenguaje de programación.*

-Hardware

Consiste en una placa PCB (printed circuit board) (ver figura), es decir, un circuito impreso compuesto de un microcontrolador y una serie de patillas hembra que pueden funcionar tanto como entradas, como de salidas.

Existen numerosos tipos de placas, cada una con unas determinadas dimensiones, un modelo de microcontrolador, requerimientos de alimentación, etc. En este proyecto utilizaremos la placa Arduino UNO, que para una información detallada puede consultarse el datasheet de los anexos.

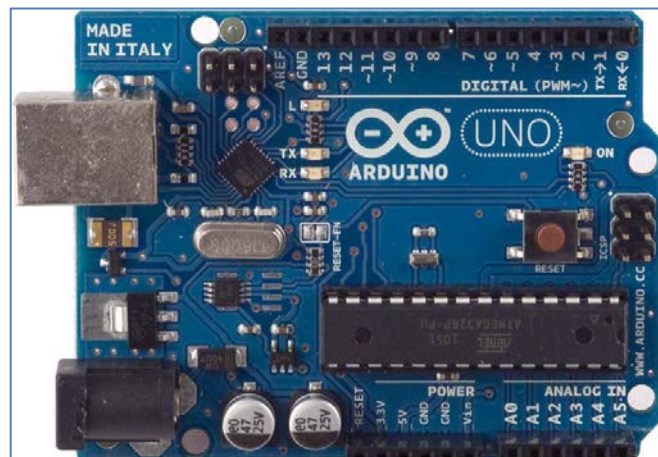


Fig. 2.2. Placa Arduino

Para abordar la alimentación se pueden emplear *dos métodos* mediante los cuales se puede mandar tensión al Arduino:



- El primero consiste en una conexión USB con el PC, que aparte de para la comunicación (como más tarde explicaremos) también la podemos utilizar para alimentar la placa.
- El segundo consiste en emplear una fuente externa con una tensión de 7-12v.

Los pines hembra de **POWER**, que nos proporcionan salidas de tensión continua, son los indicados en el recuadro rojo indicado en la figura 2.3. Sus funciones son las siguientes:

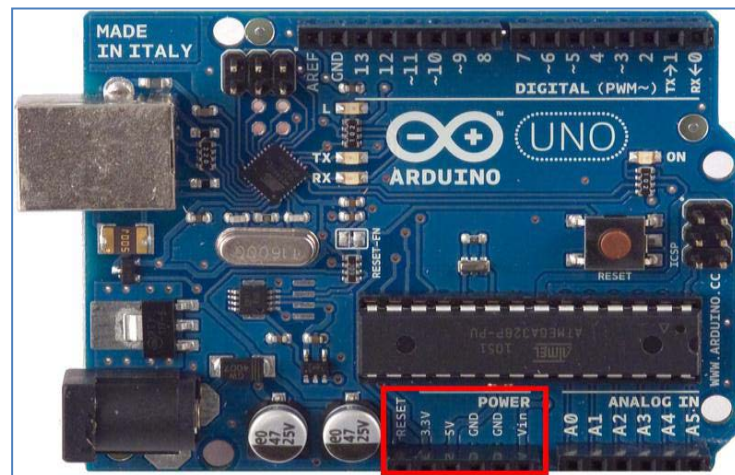


Fig. 2.3-Pines de alimentación de Arduino

- **V in**

Se obtiene la tensión a la que alimentamos la placa con la fuente externa.

- **GND**

Son las tomas a tierra. Aparecen dos.

- **3.3V**

Como indica su propio nombre, proporciona 3.3v a 50 mA mediante un regulador interno.

- **5V**

También revela que se alcanzan 5v.

Por otra parte tenemos 14 pines (ver figura 2.4) denominado **digital Inputs/Outputs**, que se usan como entradas o salidas digitales y pueden

recibir u otorgar un máximo de 40mA y 5v. A estos pines-hembra se les denomina *GPIO* “General Purpose Input/Output”.

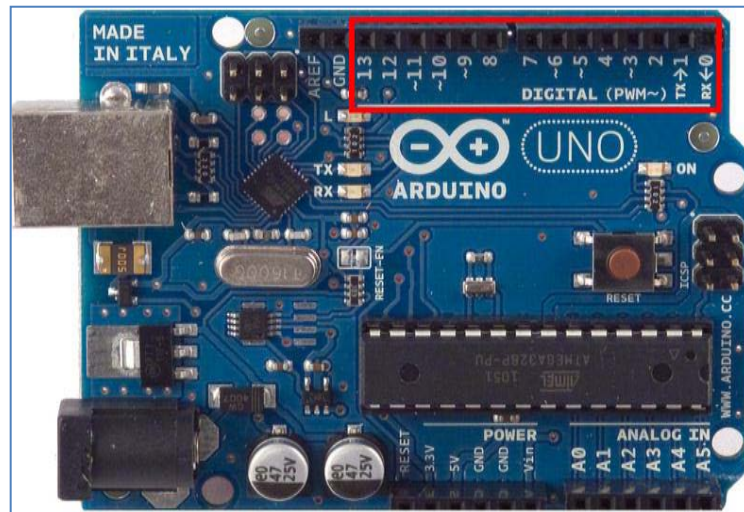


Fig. 2.4-Pines digitales de Arduino

Cada uno de ellos tiene funciones concretas:

- **Pin 0 y 1**

El “0” se denomina *RX* y se utiliza para recibir bits. El “1” se denomina *TX* y transmite bits en serie.

- **Pin 2 y pin3**

Se utilizarán como interrupciones en el programa secuencial tras algún evento externo.

- **Pin 3, 5, 6, 9, 10 y 11**

Existe una controversia puesto que al utilizarse para la función *PWM* (modulación por ancho de pulso); puesto que es una simulación de una salida analógica, pero utiliza salidas digitales para crearla.

- **Pines 10, 11, 12 , 13**

Son pines de apoyo a la comunicación *SPI* (serial peripheral Interface). *SPI* es un sistema de comunicación estándar que controla a pequeñas distancias prácticamente cualquier dispositivo digital, que permita aceptar un flujo de bits en serie y sincronizados.

Así mismo, dispone de unas entradas analógicas o *Analog Inputs* (ver recuadros de la siguiente imagen).

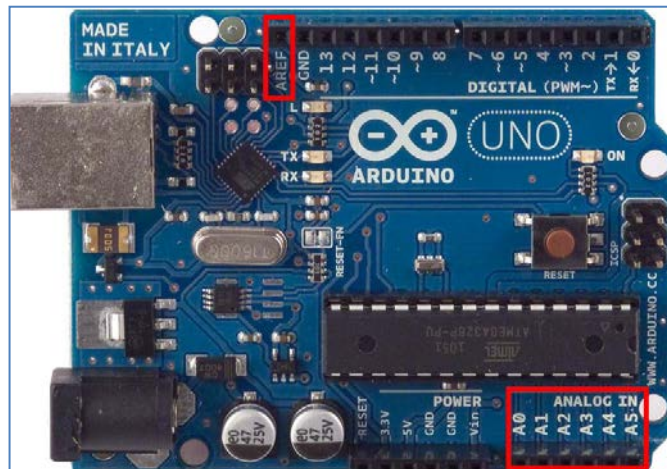


Fig. 2.5- Pines analógicos de Arduino

Como se observa en la figura 2.5 se denominan entradas analógicas desde **A0 a A5**, cuya resolución es de 10 bits (1024 estados) con una tensión de 5v variable.

AREF es un pin que nos proporciona la señal que introduzcamos desde el exterior.

Por su parte, podemos clasificar conjuntos de pines en los siguientes puertos:

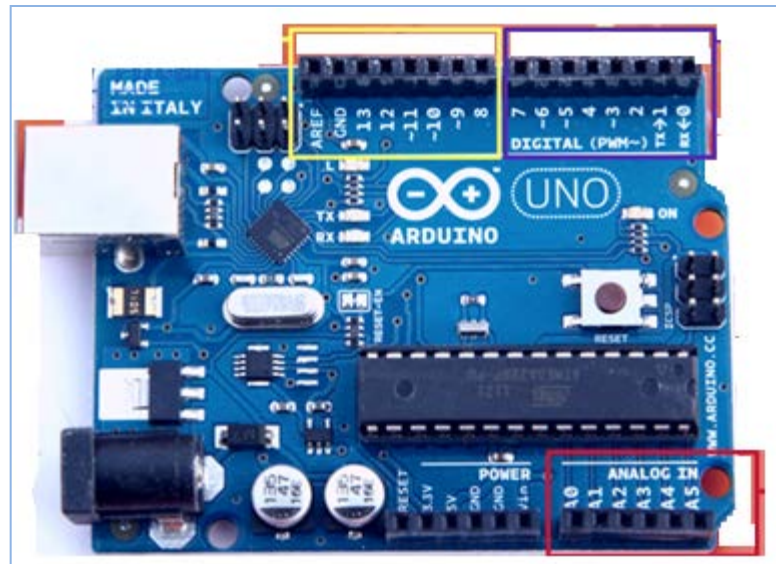


Fig. 2.6. Puertos de la placa Arduino



- **PORT B**

Pines digitales del 8 al 13 (recuadro de color amarillo en la imagen).

- **PORT C**

Pines analógicos de entrada (recuadro de color rojo en la imagen).

- **PORT D**

Pines digitales del 0 al 7 (recuadro de color morado en la imagen).

-Software

Es el entorno de desarrollo que nos permite escribir, verificar y cargar el programa que realicemos en la memoria del microcontrolador. Este entorno es bastante intuitivo y sencillo de utilizar. La descarga al ordenador se podrá llevar a cabo a través de Arduino que, como norma general, no genera coste alguno.

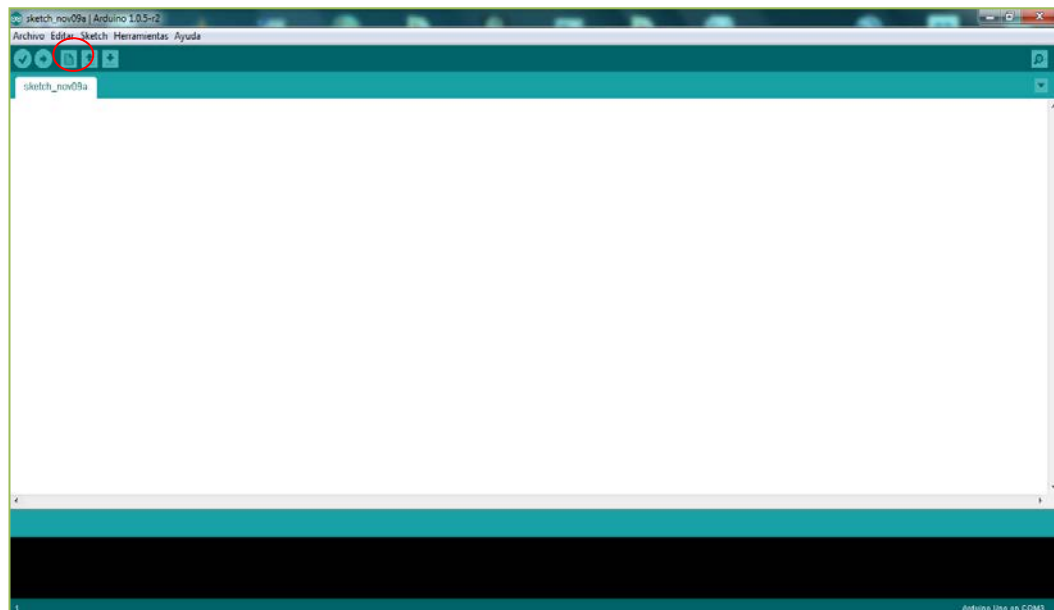


Fig.2.7- Pantalla de inicio software Arduino

En la imagen se observa el programa nada más abrirlo. En la imagen aparecen en círculos de color rojo las principales opciones que vamos utilizar. El “tick” sirve para verificar un programa creado, es decir, compilarlo para detectar posibles fallos. El icono de la “flecha” sirve para transmitir el programa a la placa.

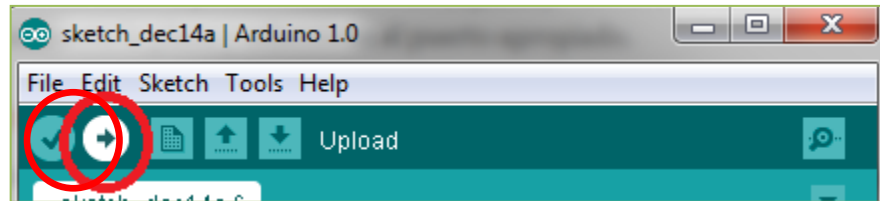


Fig. 2.8-Herramientas del software

La comunicación entre la placa y el ordenador se realiza mediante un cable USB. Si el proyecto que se realice es autónomo (el programa no varía), no será necesario mantener la conexión con el computador, por el contrario, cuando no lo sea, se deberá mantener dicha conexión.



Fig. 2.9- Cable USB para conexión placa Arduino-PC

Lenguaje de programación

Es el conjunto de instrucciones que conforman el idioma para trabajar con Arduino. Es un lenguaje de muy alto nivel y su sintaxis es muy parecida al “C”.

Un *sketch* o programa de Arduino que se va a ejecutar consta siempre de las siguientes secciones:

- Declaración de **variables globales**
- **Void setup()**

Las instrucciones situadas en esta sección se ejecutan una sola vez. Siempre al comienzo de ejecutar el programa o al resetear la placa.



- **Void loop()**

Las instrucciones dentro de esta sección se ejecutan infinitas veces hasta que se apague la placa o al resetearla.

Hay que tener en cuenta detalles como que en la sintaxis del programa se distinguen *mayúsculas* de *minúsculas*.

Sin embargo las *tabulaciones* no quieren decir nada, simplemente se utilizan para clarificar al programador si se encuentra en una de las secciones antes citadas.

Es de especial relevancia cerciorarse de acabar todas las instrucciones (incluyendo declaraciones de variables) *con punto y coma*. Es indispensable ponerlo ya que sirve para indicar que finaliza dicha instrucción.

Los comentarios, que resultan muy útiles a la hora de explicar un programa a un lector lo vea, se pueden marcar de dos maneras. O bien se utiliza el símbolo `// comentario` o bien se coloca entre símbolos `/* comentario */`.

Por otro lado las **variables** son elementos con un determinado contenido. Son imprescindibles para el funcionamiento programas. Hay dos tipos de variable; la variable **local**, que es aquella que se encuentra en el `void setup()` o en el `void loop ()`. Por el contrario, si se encuentra fuera de estas secciones, normalmente al principio hablamos de una variable **global**.

Las variables locales solo se pueden utilizar en las secciones correspondientes del sketch; al contrario que las globales que las podremos utilizar en cualquier parte del programa.

Para declarar un variable se puede utilizar la sintaxis: *tipoVariable nombreVariable*. El *nombreVariable* puede ser el que nosotros queramos, se suele usar uno representativo para de un vistazo saber a que hace referencia. Por el contrario, el *tipoVariable* nos indica que tipo de datos almacena, los tipos más representativos son:

- **Boolean**

Puede almacenar dos tipos de valores, cierto o falso.

- **Char**

La variable sólo puede tener un carácter (letra, dígito, signo de puntuación...)



- **Int**

El valor que contiene esta variable tiene un rango de (-2^{15}) y $(2^{15}-1)$.

Hay otros *tipos* tales como, **byte**, **word**, **short**, **float**, **double**, **array**, **long**, **unsigned long**, etc., que pueden consultarse más al detalle en la documentación de la bibliografía si el lector tiene curiosidad. En este trabajo utilizaremos **int** ya que se ajusta a nuestros requerimientos.

A continuación se describen los **comandos** principales que utilizaremos puesto hay una gran diversidad de los mismos:

- **PinMode()**

Si vamos a utilizar pines digitales del módulo de Arduino, es imprescindible definir estos pines en la sección *void setup()* como entradas o salidas. El primer parámetro será el número de pin al que nos referimos. El segundo parámetro servirá para indicar si se trata de una entrada INPUT o una salida OUTPUT.

- **DigitalWrite()**

Envía un 0 lógico LOW o un 1 lógico HIGH (segundo parámetro) al pin señalado (primer parámetro). La señal es tipo *int*.

- **DigitalRead()**

Nos devuelve el valor leído del pin, indicado como parámetro. Nuestros pines analógicos serán, del A0 al A5.

- **AnalogWrite()**

En este caso, lo que envía es una señal tipo byte (segundo parámetro). Esta señal representa el ciclo de servicio del PWM. Como ya indicamos los pines que permiten utilizar el PWM son 3, 5, 6, 9, 10 y 11 (primer parámetro).

La señal PWM la entendemos como una señal digital cuadrada, la cual utilizamos para simular una señal analógica. Tiene una resolución de 28 bits, es decir 256 valores diferentes. De esta manera si en el segundo parámetro del comando `analogWrite` introducimos un 0, significa que el pulso no dura nada y equivale a 0 voltios.



Por el contrario, si escribimos 255, será el mayor valor ciclo de servicio posible, los 5 voltios que nos proporciona Arduino. Con el valor intermedio 125, equivaldría a una señal de 2.5 voltios. Variando este parámetro seremos capaces de modificar el ancho de pulso, y por tanto la velocidad del motor.

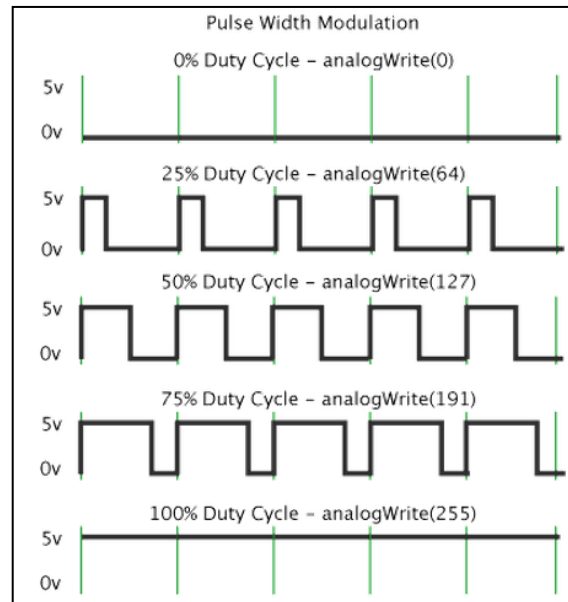


Fig. 2.10-Variación de duración de ancho de pulso

- **Delay()**

Consiste en pausar el sketch durante la cantidad de milisegundos que se ponga dentro del paréntesis. No tiene valor de retorno.

- **Map()**

Utilizaremos esta función para adecuar la señal obtenida de los sensores a un rango de trabajo. Es decir, modifica un valor, que será nuestro primer parámetro. Este primer parámetro está dentro de un rango; el mínimo del rango será el segundo parámetro y el máximo el tercero. Si ponemos un cuarto y un quinto parámetro serán los mínimos y máximos respectivamente de otro rango; el cual sirve para que el cambio de valor se realice más proporcionalmente.



- **If(){ }/else (){ }**

Sirve para verificar que se cumple una condición; que si es cierta se ejecuta el código que hay entre los corchetes del *if*. Si es falsa se ejecutaría el código situado entre los corchetes del *else*.

- **Switch()**

Realiza básicamente la misma función que el anterior comando, pero de una manera más elegante. Básicamente consiste en elegir entre varios casos (*case*) que pueden darse. Al final de cada caso se requiere escribir un *break* indicando que el susodicho caso ha terminado.

Los **operadores aritméticos** son los siguientes:

- **+** operador suma
- **-** operador resta
- ***** operador multiplicación
- **/** operador división
- **%** operador módulo

Por su parte los **operadores lógicos** son:

- **&&** operador AND (las dos condiciones son correctas)
- **||** operador OR (al menos una de las condiciones es correcta)
- **!** operador NOT (comprueba que no se cumpla la condición)
- **&** operador AND bit a bit
- **|** operador OR bit a bit
- **~** operador NOT bit a bit

También aparecen **operadores de comparación**:

- **==** compara igualdad
- **!=** compara diferencia
- **>...** compara mayor que
- **>=** compara mayor o igual que
- **<...** compara menor que
- **<=** compara menor o igual que

Es importante hacer mención al elemento **TCCRnB** o *timer* y que se utiliza para cambiar la frecuencia de funcionamiento del PWM.



La letra **n** hace referencia al número de *timer* utilizado; (por ejemplo: TCCR2B será el timer **2**). Hay 4 timers y dependiendo de cuál utilicemos controlaremos unas patillas u otras:

- **Timer0**: controla las patillas 3 y 4.
- **Timer1**: controla las patillas 12 y 11.
- **Timer2**: controla las patillas 10 y 9.
- **Timer3**: controla las patillas 5, 3 y 2.
- **Timer4**: controla las patillas 8, 7 y 6.

El TCCRnB consta de 8bits. Los 3 primeros, derecha a izquierda de denominan CS02, CS01, CS00. Estos tres bits representan el denominado *prescaler*, el cuál al tener 3 dígitos su rango es de 0-7 en decimal. El prescaler es lo que hay que cambiar si queremos cambiar la frecuencia. Pero tenemos que tener en cuenta que no todas las patillas funcionan a la misma frecuencia. Las patillas que vamos a controlar (9, 10, 11) pueden trabar a estas frecuencias:

10-9

Setting	Divisor	Frequency
0x01	1	31372.55
0x02	8	3921.16
0x03	64	490.20 <--DEFAULT
0x04	256	122.55
0x05	1024	30.64

11-3

Setting	Divisor	Frequency
0x01	1	31372.55
0x02	8	3921.16
0x03	32	980.39
0x04	64	490.20 <--DEFAULT



0x05	128	245.10
0x06	256	122.55
0x07	1024	30.64

Con todos estos conocimientos sobre Arduino seríamos capaces de programar el sistema de control del motor.

Para este trabajo he elegido el controlador Arduino frente a otros, como por ejemplo el “PIC”, por una serie de ventajas tales como:

- Tiene un hardware y un software **libre** y **extensible**. Esto quiere decir que los podemos variar, mejorar, y adaptar a nuestras necesidades sin problemas, es decir, son **versátiles**.

Según la *Free Software Foundation* para que se considere libre debe poseer las siguientes características:

- Poder usar el programa con cualquier propósito.
 - Permitir estudiar el funcionamiento interno del programa y adaptarlo.
 - Distribuir copias a voluntad.
 - Mejorar el código fuente y poder difundirlo públicamente para un beneficio global.
- Además el software también es **multiplataforma** ya que lo podemos instalar en sistemas operativos como Linux, Windows o Mac OS.
 - El **bajo coste** de sus placas lo hace atractivo para emprender en la realización de proyectos que se realizan por hobby o para pequeñas aplicaciones. Si añadimos que las placas son **reutilizables** para realizar otros proyectos; podemos usarlo indefinidamente.
 - Por último remarcar la **sencillez** y **claridad** de su lenguaje de programación. Cuando decimos que es sencillo, nos referimos a personas que ya han programado en otro lenguaje; no les hace falta mucho esfuerzo ni conocimientos a mayores para poder programar el Arduino.

2.2. CONVERTIDOR DC/AC MODELO L6234

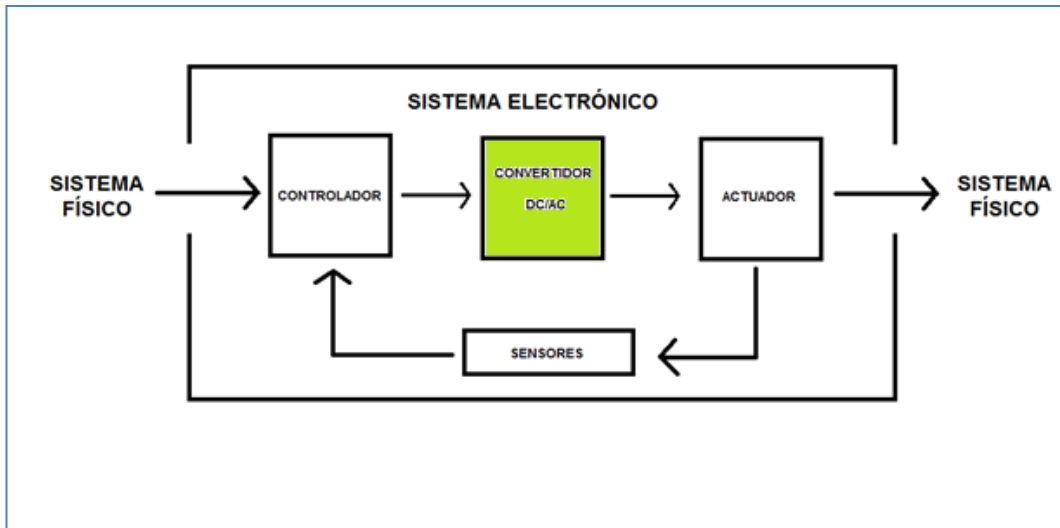


Fig. 2.11-Posición del convertidor DC/AC en el sist. electrónico

El *convertidor DC/AC*, también denominado **inversor**, es el encargado de transformar una señal que le llega DC en una AC, ya que por conmutación genera una secuencia de salida con magnitud, frecuencia y fase variable y controlable.

Se pueden clasificar según múltiples variables:

- Según la *fuentes de entrada* al convertidor.
- En función del número de niveles de salida.
- Teniendo en cuenta la topología de la etapa de potencia.
- Por las fases de salida que genera:
 - Monofásicos.
 - Trifásicos.
- Dependiendo de la *técnica de control empleada*:
 - Modulación SVM.
 - Modulación Sliding.
 - Modulación PWM.

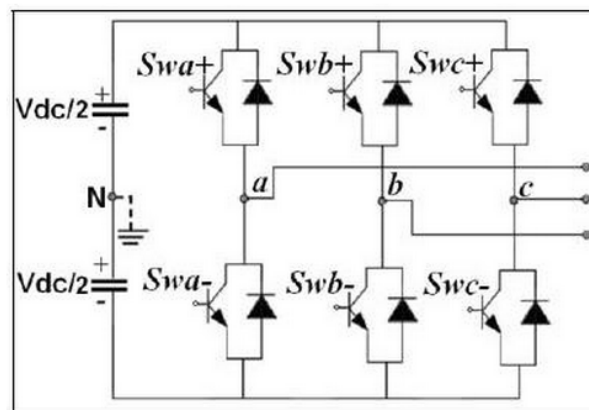


Fig. 2.12- Esquema de un inversor genérico

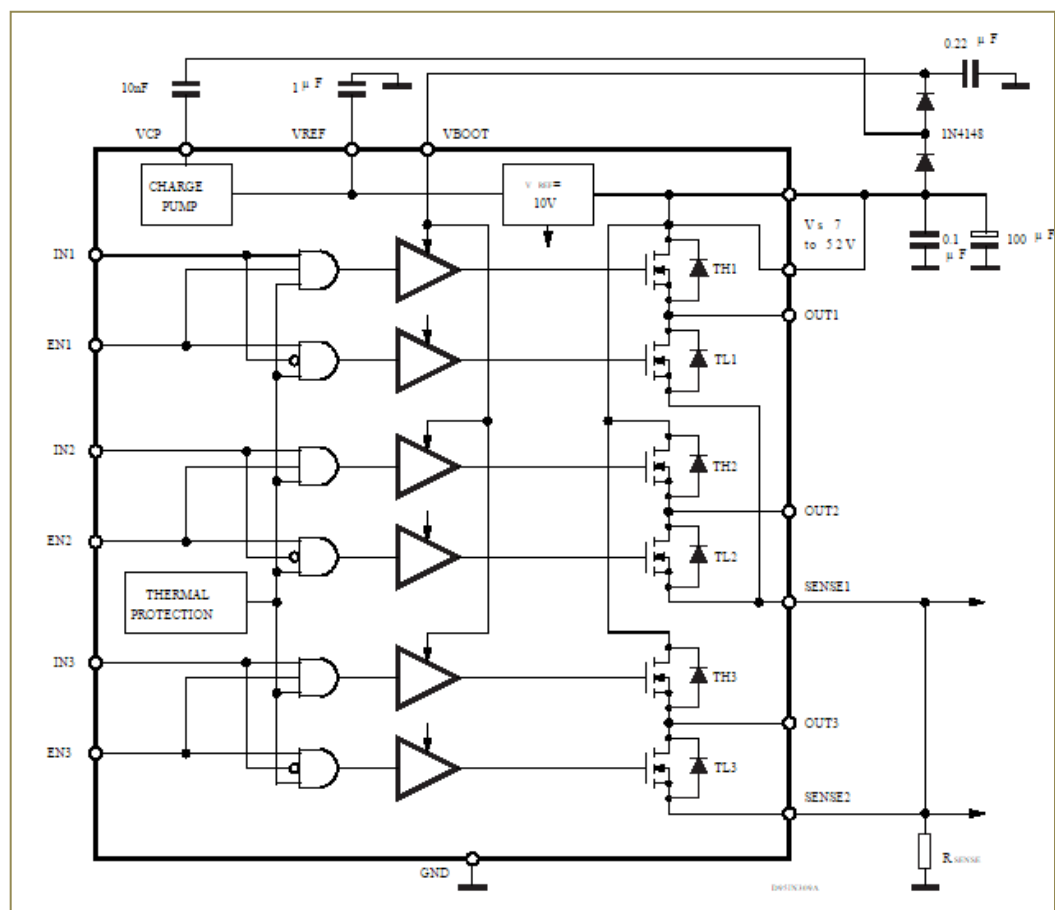
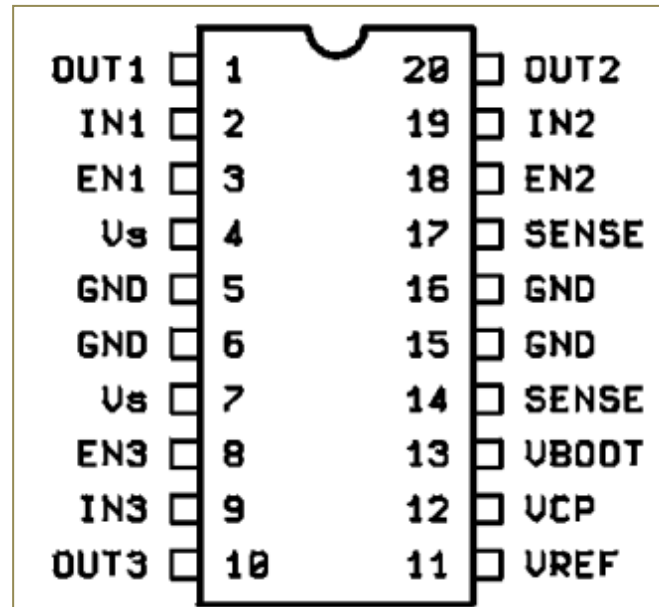
En la figura 2.12 se puede observar una configuración trifásica. Está compuesta por 6 transistores que actuarán de interruptores para generar la señal PWM. Los diodos se utilizan tanto de protección como para que haya una circulación bidireccional de corriente.

En el proyecto se va a utilizar un inversor integrado, el L6234 cuyo aspecto externo es el de la imagen 2.13.



Fig. 2.13-Inversor integrado L6234

La estructura interna del L6234 es la siguiente:



Podemos dividir en tres sectores el inversor:

Puertas AND

Permite elegir que transistores activar según la secuencia de bits. Unos párrafos indicamos estas combinaciones.

– Drivers

Disparan al transistor que indica su correspondiente puerta AND. Es interesante observar el datasheet (se encuentra en el anexo) ya que hay que tener en cuenta algún detalle como por ejemplo el *BOOTSTRAP*. El bootstrap es una técnica utilizada para no tener que utilizar una alimentación para cada transistor de la parte superior del inversor, mas otra añadida que alimente los tres inferiores. Conseguimos utilizar la misma alimentación para todos los transistores. En el esquema, el pin Vboot va precedido por dos diodos 1N4148 que son de alta velocidad; nosotros utilizaremos unos diodos schottky que son aún más rápidos. Por este pin se inyecta el voltaje necesario para activar los drivers y que alimenten la siguiente parte; los transistores.

– Transistores DMOS

Cada uno con su diodo; que ya explicamos para que servía este último. EL transistor **TL** se activa si el driver envía un 0, sin embargo si lo que envía es un 1 se activará **TH**.

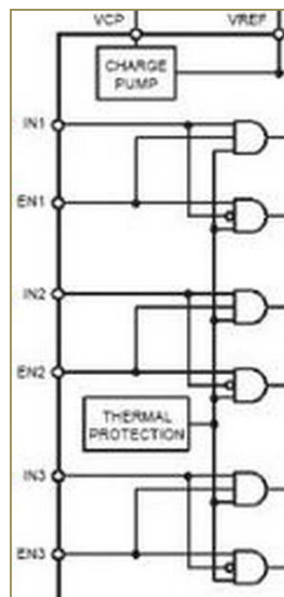


Fig. 2.16- puertas And del L6234

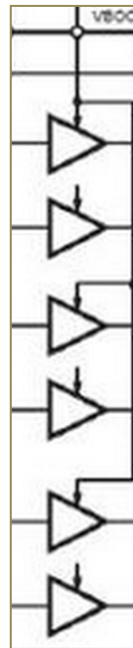


Fig. 2.17- drivers del L6234

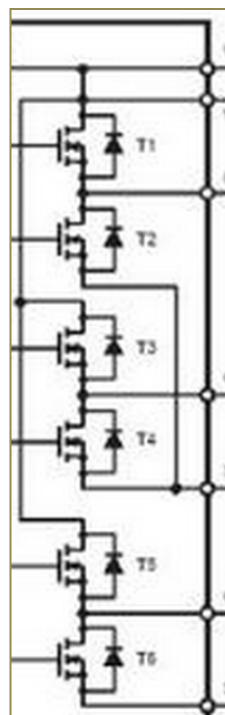


Fig. 2.18- transistores DMOS del L6234

A las salidas **OUT** del convertidor DC/AC irán conectadas las fases del motor BLDC.



Se puede facilitar el entendimiento del circuito a través de su tabla de la verdad:

THERM_PROTEC	EN	IN	OUT
0	X	X	0
1	0	X	0
1	1	0	0
1	1	1	1

Tabla. 2.1- Tabla de la verdad del inversor L6234

En el momento que la señal de **THERMAL PROTECTION** se pone a 0, el sistema deja de enviar señales a las salidas por seguridad y las mantiene a nivel bajo.

Por otra parte las entradas **EN** (*enable*) habilitan, junto con la entrada térmica la puerta AND que se vaya a utilizar. Hay que tener en cuenta, según las especificaciones que nos marca el datasheet que pueden conmutar a un máximo de 50kHz.

Por último con las entradas **IN** (*input*) elegimos si queremos que a la salida que le corresponde aparezca un 0 o un 1. La frecuencia de conmutación debería ser mayor que 16 kHz para evitar ruido.

Considerando que es de suma importancia conocer cómo opera el convertidor para poder trabajar con él, se muestra a continuación un ejemplo con la rama 2.

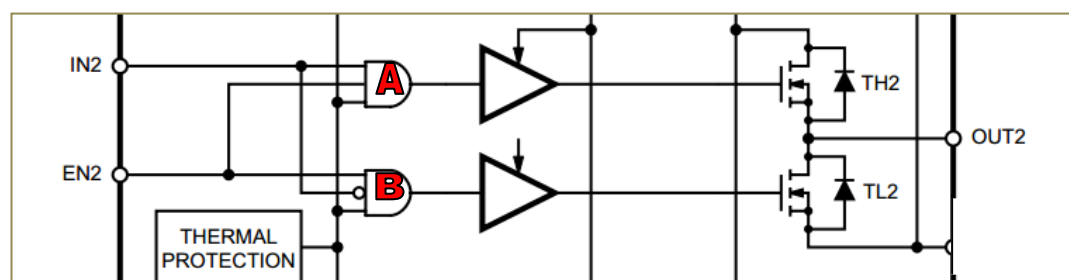


Fig. 2.19- Rama 2, de las tres que posee el inversor

Lo primero a tener en cuenta es que, a no ser que se sobrecaliente el microchip L6234, la señal Thermal Protection se mantendrá a 1. Por otro lado para habilitar ambas puertas AND, la entrada EN2 la pondremos nosotros a 1 también. Ahora bien, dependiendo si queremos en la salida OUT2 un 0 o un 1

veamos que ocurre. Para obtener un 1 en OUT2 introducimos 1 en IN2, la puerta AND A con combinación 111 emite un 1 al driver correspondiente y activa TH2 ; mientras que la puerta AND B con combinación 110 (ya que la entrada IN2 está negada) emite un 0 y el driver no actúa, no activando TL2. Como TH2 está conectado a alimentación, en OUT 2 aparece un 1 lógico.

Por el contrario, manteniendo el 1 en la entrada térmica y EN2, al introducir un 0 en IN2, la puerta AND B tiene combinación 111, activando su driver y por tanto TL2; que al ir este transistor a masa, hace que en OUT2 aparezca un 0 lógico.

2.3. MOTOR BLDC

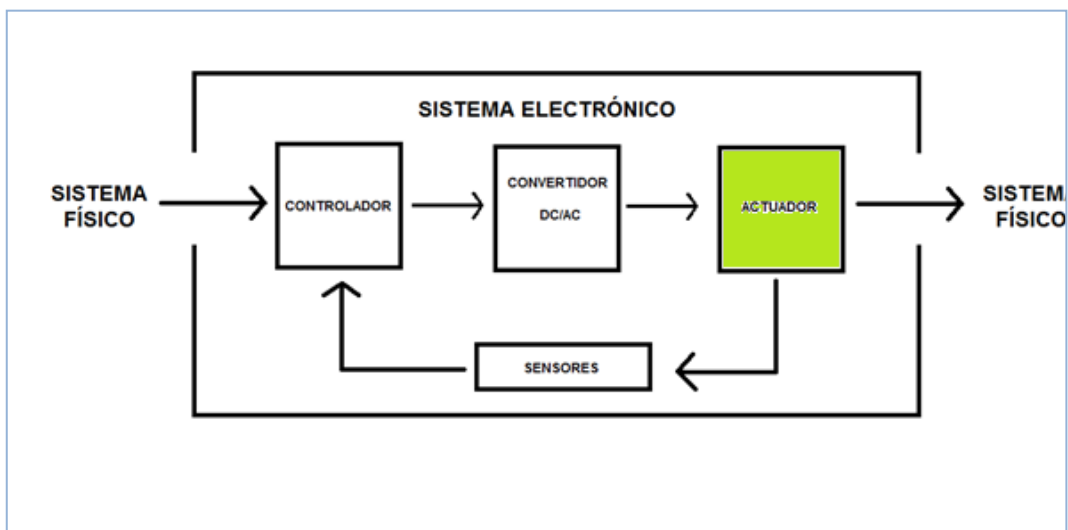


Fig. 2.20- Puesto que ocupa el motor en el sistema electrónico

Un **motor** es una *máquina eléctrica*, la cuál será el actuador del proyecto, es decir, la parte del sistema electrónico que interactuará con el mundo físico.

Una **máquina eléctrica** es básicamente una máquina rotatoria que transforma la energía eléctrica en energía mecánica y viceversa. Esta máquina se denomina *motor* si transforma únicamente la eléctrica en mecánica. Si la transformación de energía se realiza en sentido contrario, se denomina *generador* (ver fig. 21).

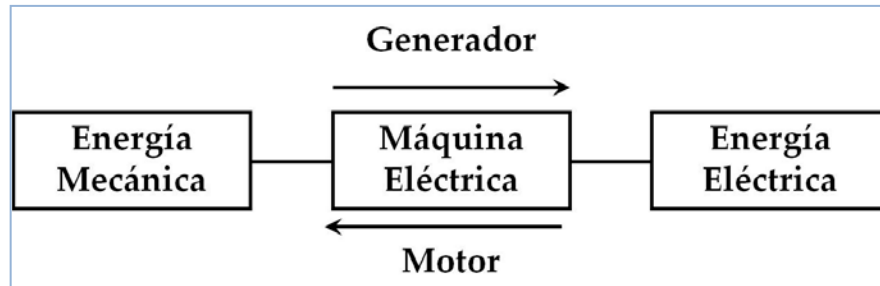


Fig. 2.21- Tipo de máquina eléctrica en función del sentido de la corriente

El funcionamiento general de los motores eléctricos está fundamentado en la atracción y repulsión entre un imán y una bobina, por la que se hace circular una corriente eléctrica.

Los componentes básicos de estos motores son por tanto bobinas e imanes; que pueden estar situados, dependiendo el modelo del motor, en los denominados *rotor* o *estator*.

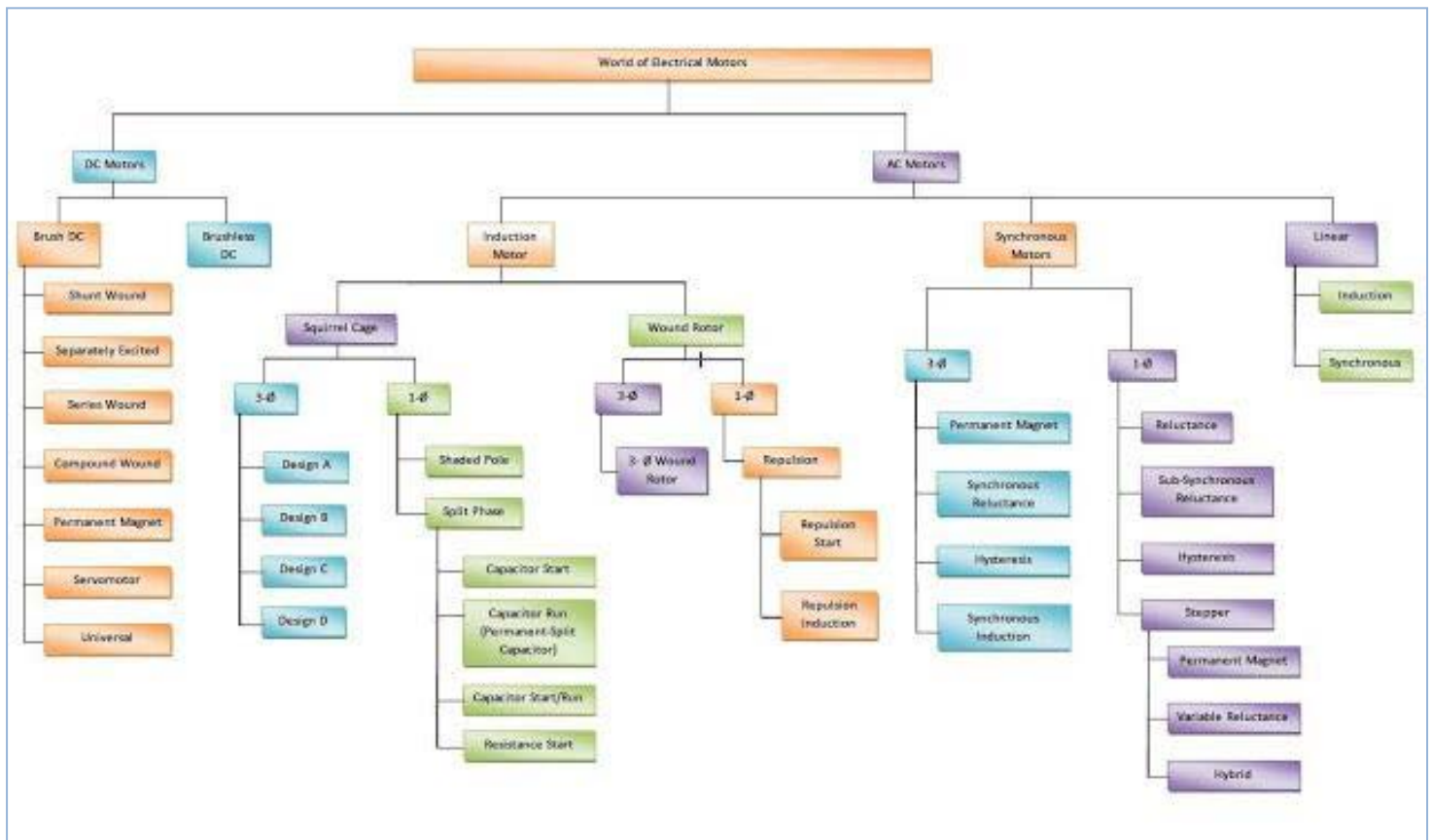


Fig. 2.22- Esquema tipos de motores



Los motores eléctricos se pueden clasificar principalmente en:

- Motores de corriente alterna (**AC**).
- Motores de corriente continua (**DC**).

Dentro de estos tipos hay multitud de variantes como se observa en el esquema anterior (fig. 22).

Describiremos brevemente las características de los dos tipos principales (AC,DC), y nos centraremos dentro del grupo DC en el modelo BLDC.

○ Motores AC

Estos motores para funcionar necesitan una fuente con tres fases. El rotor recibe una corriente alterna y el estator posee un campo inducido. Los motores de AC se utilizan para exigencias de velocidad fija, y además no pueden operar a bajas velocidades.

La eficiencia de estos motores va aumentando y se prevé que supere a los DC. Además son más baratos y fáciles de mantener que los de corriente continua.

Sin embargo son más difíciles de controlar que los de corriente continua. Por ello suelen ir equipados con una unidad de frecuencia variable. Los motores de corriente alterna de uso común hoy en día se pueden dividir en tres grandes categorías:

- Motores de inducción (asíncronos).
- Motores síncronos.
- Motores lineales.

○ Motores DC

Son utilizados en la industria del plástico, papel, acero, **automóviles...etc.**

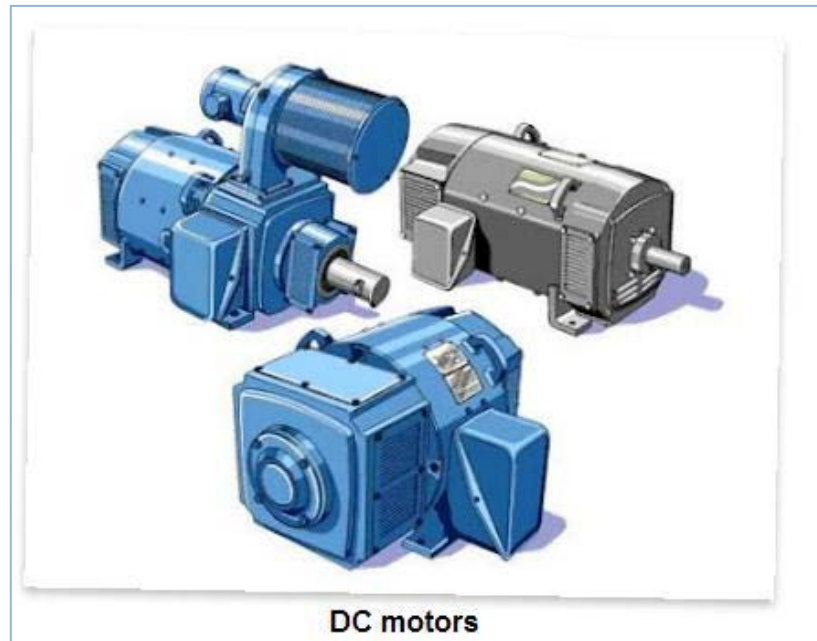


Fig. 2.23-Ejemplos de motores DC

Los motores de corriente continua están compuestos fundamentalmente por una armadura, conmutador, campo y escobillas (los motores brushless carecen de estas últimas).

- **La armadura**

Es la parte principal del motor, gira sobre dos cojinetes creando par. La rotación es producida por una corriente en los arrollamientos de las bobinas. Se muestra un ejemplo en la figura 2.24.

- **El conmutador**

Consiste en un conjunto de segmentos de cobre fijados alrededor de la circunferencia de la armadura. Como se puede comprobar en la figura 2.25 Cada segmento debe estar separado de los otros, además cada contacto actúa como un interruptor; ya que la corriente fluye a través de sólo el bobinado de la armadura con la que las escobillas están en contacto. Este flujo de corriente crea fuerza electromotriz, que a su vez es la responsable de crear el par.

- **El campo**

Es el conjunto de imanes permanentes, también conocidos como polos de campo. Estos polos son curvados para adaptarse a la

forma de la armadura. Es el que induce el campo magnético a la armadura. La imagen 2.26 ilustra la forma del campo.

- Las **escobillas** (si las tiene)

Es la parte del motor que está compuesta de un material conductor, similar al carbón colocada en la “cola del motor”.



Fig. 2.24-Armadura de un motor DC

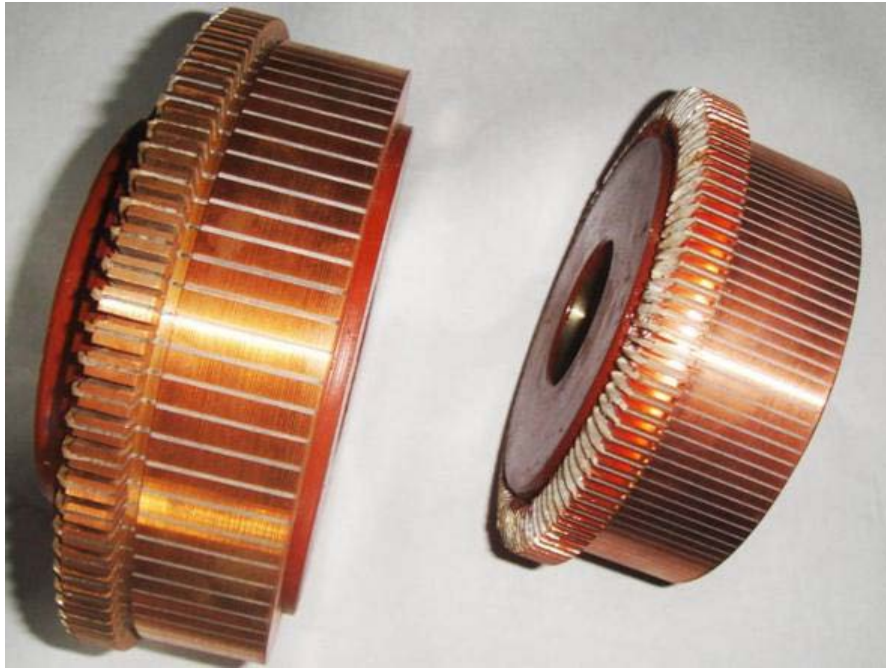


Fig. 2.25-Conmutador de un motor DC



Fig. 2.26-Campo de un motor DC



La principal característica que aventaja a estos motores es que es fácil de controlar su velocidad en una amplia gama; su característica par-velocidad ha sido, históricamente, más fácil a medida que la de todas las categorías de motores de AC. Esta es la razón por la que la mayoría de los motores de tracción y servomotores han sido máquinas de corriente continua. Por ejemplo, los motores para la conducción de vehículos ferroviarios eran, hasta hace poco, exclusivamente máquinas de corriente continua.

Por el contrario, tienen el problema de que es más cara su construcción.

Los motores DC se subdividen en:

- a) Brush DC.
- b) Brushless DC (BLDC).

a). Motor Brush DC

Para variar la velocidad en este tipo de motores es necesario cambiar la tensión que se le aplica o variar la fuerza del campo magnético.

Presenta las siguientes ventajas:

- Control más simple que los AC y BLDC.
- Costes inferiores de construcción que los BLDC.

A su vez, tiene los siguientes inconvenientes:

- Necesitan cepillos para conectar el devanado del rotor. Se produce desgaste de las escobillas. Si se utiliza en los aviones, los cepillos necesitarían reemplazo después de una hora de funcionamiento.
- Las chispas de los cepillos pueden causar una explosión si el entorno contiene material explosivo.
- Ruido de RF de los cepillos puede interferir con dispositivos electrónicos.

b) Motor Brushless DC (BLDC)

Como su nombre lo indica, motores BLDC no utilizan escobillas para la conmutación. Consiste en **estator** con una serie de slots devanados y un **rotor** con una serie de polos, creados con imanes permanentes. Se utilizan en las industrias tales como electrodomésticos, discos duros, industria aeroespacial, automóviles, instrumentos médicos, equipos de automatización, etc.



Fig. 2.27-Rotor (izq) y estator (der) de un motor BLDC

El **estator** se constituye de una serie de láminas de acero apiladas con unas aperturas. En estas aperturas o ranuras, se sitúan los devanados.

El **rotor** dispone los imanes permanentes. Lo normal es que estos imanes tengan una composición de NdFeB. Es una composición de alta densidad de energía, además el neodimio es, dentro de las tierras raras la más común; por lo que se abarata el coste.

La estructura más típica de estos motores se muestra en la figura 2.28.

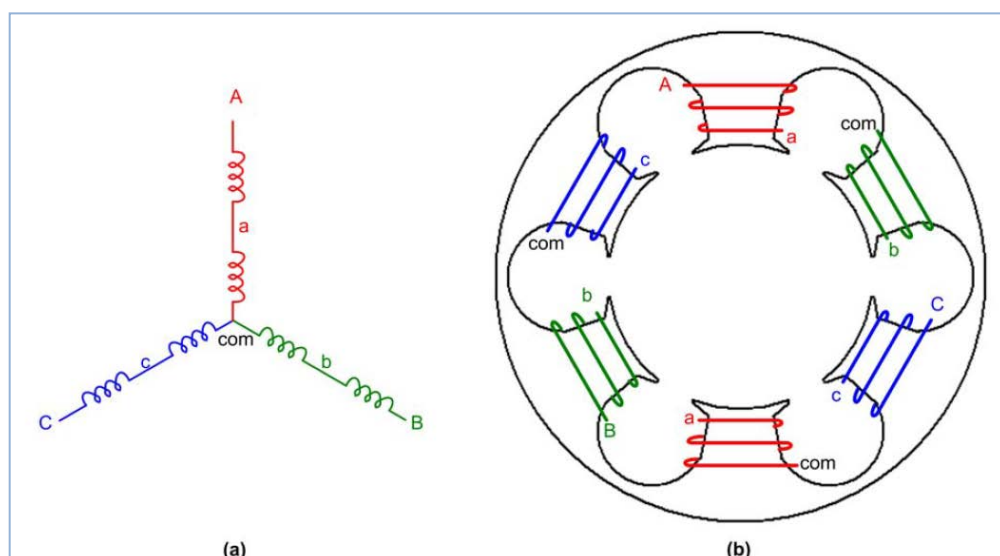


Fig. 2.28-Configuración BLDC, 3 bobinas (a), 6 polos de potencia (b)

Normalmente se componen de tres fases, las cuales, a su vez, están desfasadas 120° . Cada bobina del estator se puede subdividir en otras dos, las cuales representan dos polos de potencia.

Estas fases (A, B, C) poseen dos bobinas cada una, es decir un total de seis polos de potencia.

Para que gire el motor es necesario una conmutación electrónica en las bobinas. Se debe crear un campo magnético perpendicular a la dirección del rotor.

Dependiendo de la posición del rotor respecto del estator; en cada momento deben estar excitadas las bobinas correspondientes. Para conocer esa posición se suelen emplear sensores (se explicará en el siguiente capítulo). Aunque hay un tipo de control para motores *sensoless*, es decir, sin sensores.

Hay que buscar el par máximo que se produce al interaccionar el campo magnético generado por el estator y los imanes del rotor. Este aparece en el momento que ambos campos están a 90° uno del otro. Esto se puede comprobar con la ecuación 2.1. Puesto que el rotor se va moviendo, hay que ir cambiando la excitación de las bobinas para conseguir mantener este desfase de 90° .

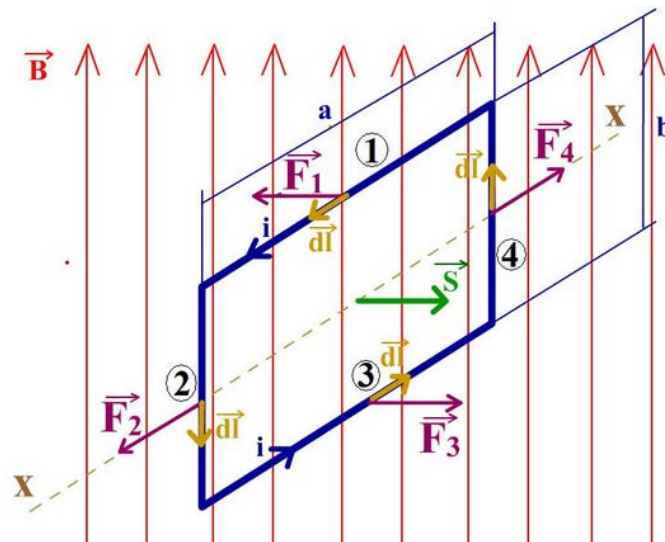


Fig. 2.29-Conjunto de fuerzas electromagnéticas que intervienen en la Fuerza de Lorentz.

$$F = L \cdot i \cdot B \cdot \sin\phi$$

Ec. 2.1- Fuerza de Lorentz.

Donde:

F = Fuerza de Lorentz [N]

L = longitud del conductor [m]

B = intensidad del campo magnético [Tesla]

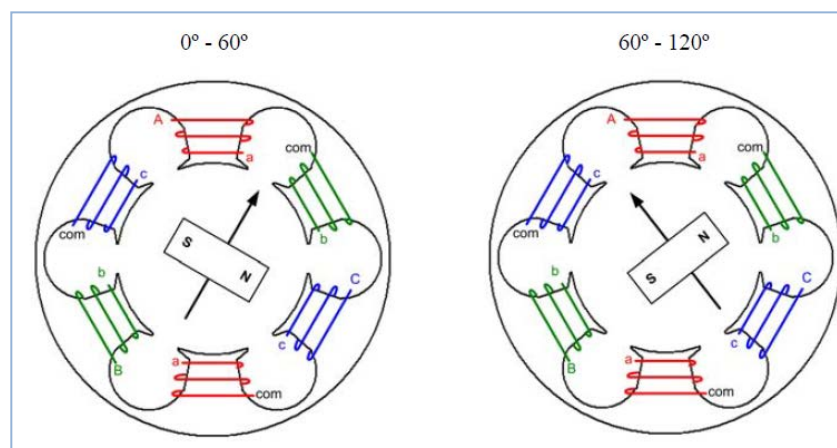
φ = ángulo entre B e I

Esta secuencia de excitación corresponde con una determinada activación de dos bobinas, de manera que una de ellas tenga tensión positiva y la otra tensión negativa. La tercera no tiene tensión (no está activada).

Para controlar el motor hay tres técnicas aplicables:

- Conmutación trapezoidal, también denominada 6 step mode.
- Conmutación sinusoidal.
- Control vectorial.

Nosotros utilizaremos la vectorial, aunque es prácticamente igual que la trapezoidal. Dividiremos los 360° de giro del motor en 6 conmutaciones de bobinas, por lo que cada 60° que gire el rotor, se deberá cambiar el vector. Se muestra el giro en la figura 2.30.



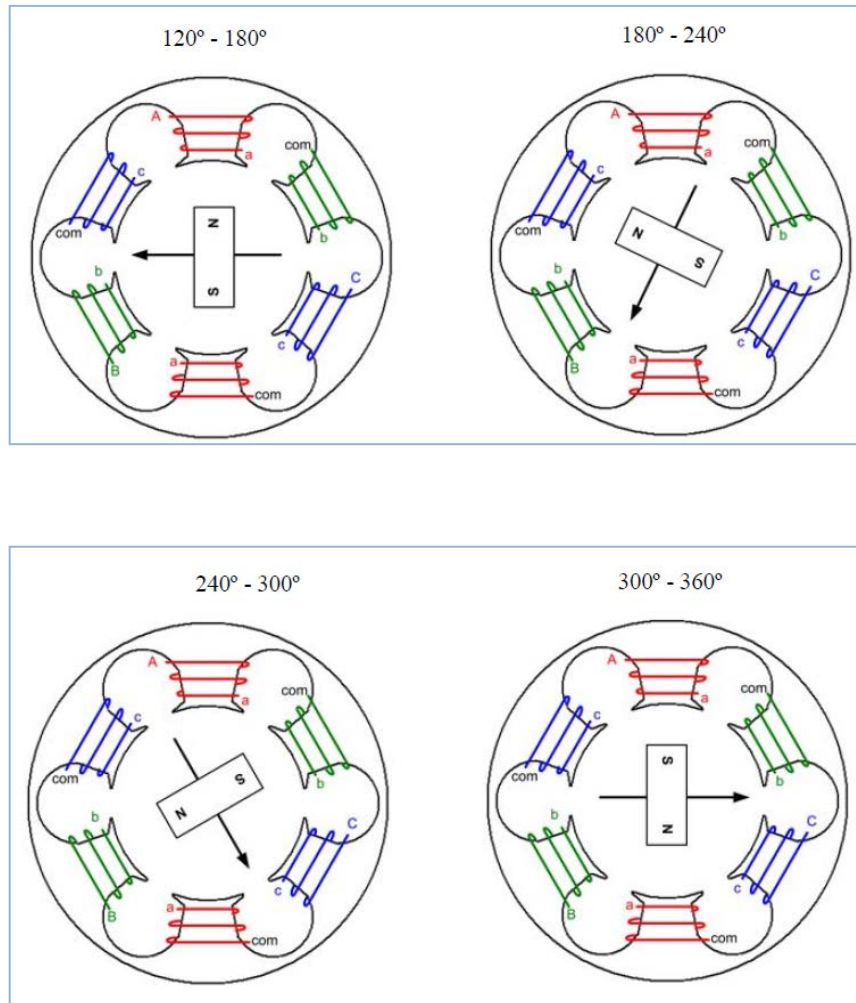


Fig. 2.30- Secuencia de giro del motor BLDC

Sus ventajas son:

- No disponen de conmutación mecánica, ello hace que sean más silenciosos.
- Proporcionan una buena respuesta dinámica. Presenta una mayor aceleración
- Los rangos de velocidad son mayores. Llegan a altas velocidades tales como 100.000 rpm.
- No producen chispas, permite trabajar en atmósferas explosivas.
- Tienen alta eficiencia (alrededor del 90%), se debe a que el motor apenas se recalienta en su conversión de energía eléctrica a mecánica.
- La relación par motor-tamaño es mayor, lo que le permite trabajar en espacios reducidos.
- Su vida útil elevada.

A su vez, sus inconvenientes son los siguientes:

- Su control es bastante complejo.
- Su construcción tiene un mayor coste.

En la siguiente tabla⁵ se muestran las principales diferencias entre el motor BLDC y un motor Brush DC (con escobillas).

	Motor BLDC	Motor con escobillas
Conmutación	Conmutación electrónica basada en sensores de posición de efecto Hall	Conmutación por escobillas
Mantenimiento	Mínimo	Periódico
Durabilidad	Mayor	Menor
Curva Velocidad / par	Plana. Operación a todas las velocidades con la carga definida	Moderada. A altas velocidades la fricción de las escobillas se incrementa, reduciendo el par.
Eficiencia	Alta. Sin caída de tensión por las escobillas.	Moderada
Potencia de salida / Tamaño	Alta. Menor tamaño debido a mejores características térmicas porque los bobinados están en el estator, que al estar en la carcasa tiene una mejor disipación de calor.	Baja. El calor producido en la armadura es disipado en el interior aumentando la temperatura y limitando las características.
Inercia del rotor	Baja. Debido a los imanes permanentes en el rotor	Alta. Limita las características dinámicas.
Rango de velocidad	Alto. Sin limitaciones mecánicas impuestas por escobillas/conmutador.	Bajo. El límite lo imponen principalmente las escobillas.
Ruido eléctrico generado	Bajo	Arcos en las escobillas
Coste de construcción	Alto. Debido a los imanes permanentes	Bajo.
Control	Complejo y caro	Simple y barato.
Requisitos de control	Un controlador es requerido siempre para mantener el motor funcionando. El mismo puede usarse para variar la velocidad.	No se requiere control si no se requiere una variación de velocidad.

Tabla 2.2- Comparativa de motores DC con y sin escobillas

2.4. SENSORES TIPO HALL

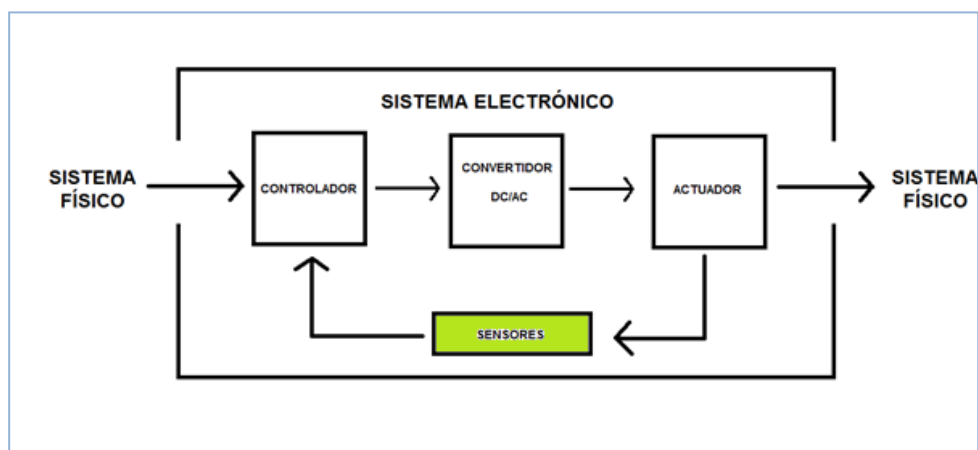


Fig. 2.31-Puesto de los sensores en el sistema electrónico

⁵ Extraída de la página "Microchip Technology In".

Los sensores son la parte que realimenta nuestro sistema electrónico ya que se encarga de devolver la información a Arduino para que este actúe de una u otra determinada forma y envíe la correcta información a la siguiente etapa.

En este trabajo, los sensores vienen integrados en nuestro motor, por lo que no será necesario instalarlos. Concretamente lleva tres sensores y su funcionamiento se basa en el principio del *efecto Hall*.

El principio del **efecto Hall** nos indica que, en un conductor por el que circula una corriente eléctrica; si se ejerce un campo magnético transversal, las cargas son impulsadas hacia un lado y otro del conductor, apareciendo un voltaje denominado *voltaje de Hall*.

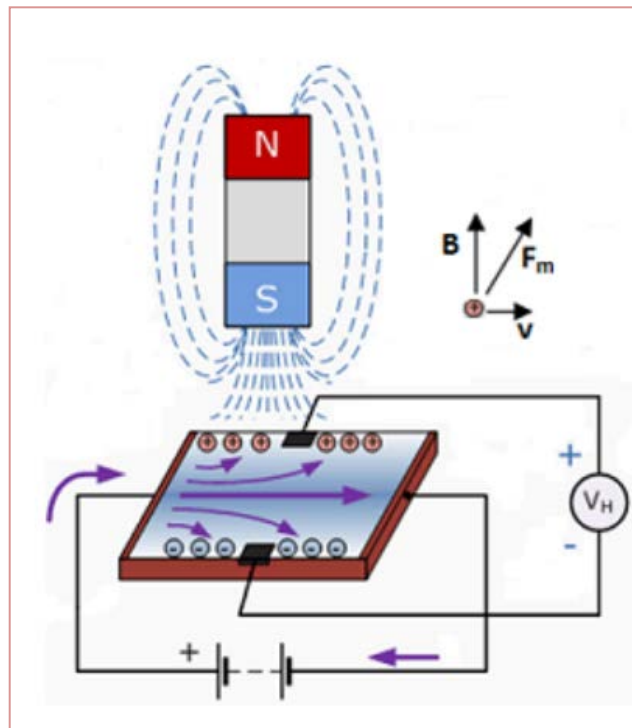


Fig. 2.32-Representación gráfica del efecto Hall

Los sensores los utilizaremos para determinar la posición del rotor, y de esta manera poder aplicar correctamente los vectores de impulsos. Al detectar el sensor el campo creado por el imán, se genera un impulso que registrará Arduino y de esta manera localiza en qué posición se sitúa el motor.

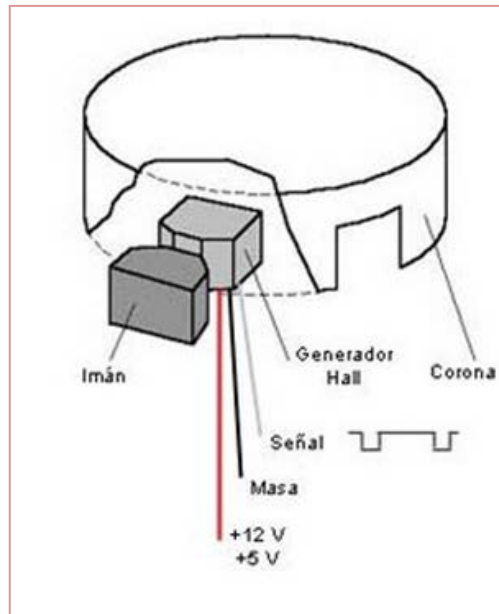


Fig. 2.33-Componentes de un sensor Hall

Un detalle muy a tener en cuenta y que no figura en el datasheet y debería estar, es que el sensor actúa como un colector abierto. Para su explicación lo compararemos a un transistor.

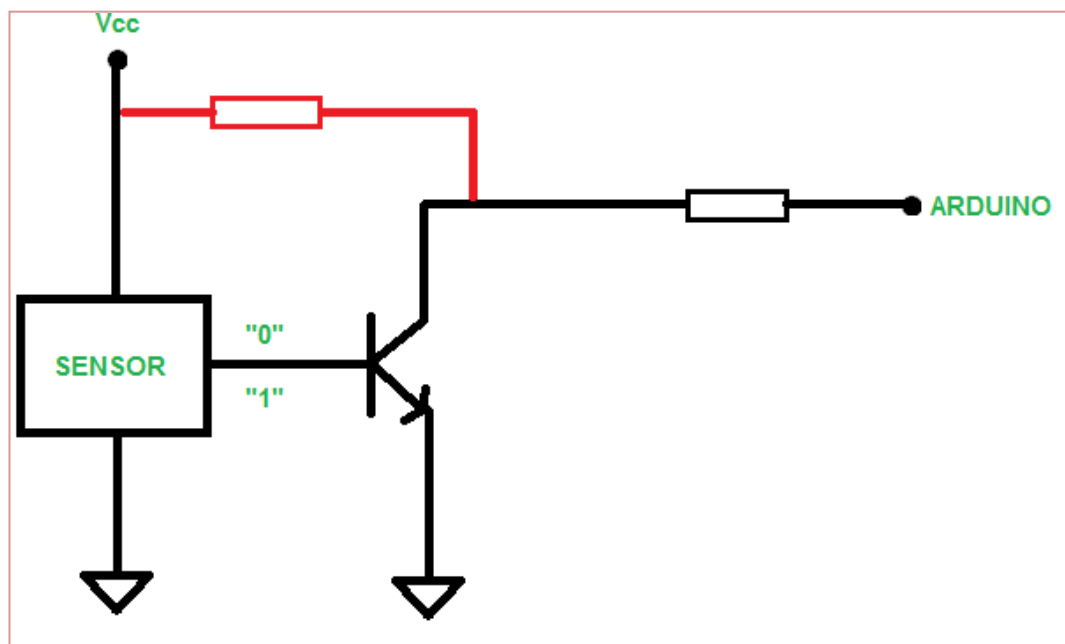


Fig. 2.34 Sensor con transistor en colector abierto



Al estar en colector común, es necesaria la polarización del transistor añadiendo una resistencia de $10K\Omega$ () representada de color rojo en la figura 2.34. De esta manera a Arduino le llegará correctamente el “0” o el “1” correspondiente del sensor.

En la figura 2.35 se muestra un ejemplo de la apariencia real de este tipo de sensor:



Fig. 2.35-Ejemplo de Sensor Hall

3

DESARROLLO DEL CONTROLADOR DE VELOCIDAD

PARA MOTOR BLDC

Tras conocer cada bloque que conforma nuestro *sistema electrónico*, seguidamente se describirá el funcionamiento del conjunto.

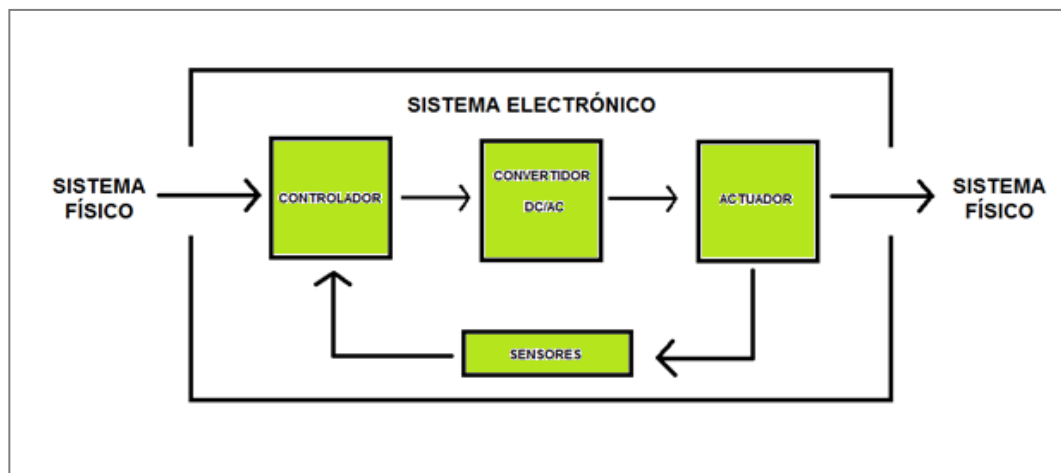


Fig. 3.1-Conjunto que conforma el sistema electrónico

El material utilizado es el siguiente:

- Motor “NT Dynamo BLDC 24v - 1Amp”.
- Placa protoboard de doble cara.
- Cables sólidos de 22AWG.
- Dos cables de alimentación.
- Placa “Arduino UNO”.
- Tres resistencias de 120 k Ω .
- Tres resistencias de 10k Ω .
- Seis resistencias de 400 Ω .
- Una resistencia de 0.1 Ω .
- Potenciómetro lineal de 100 k Ω .
- Inversor integrado “L6234”.
- Un portachip.
- Dos condensadores de 100uF.
- Un condensador de 10 nF.
- Un condensador de 220nF.
- Un condensador de 1uF.



- Un condensador de 100nF.
- Tres diodos schottky.
- Un fusible de 2.5 A.
- Un portafusible.
- Siete regletas.
- Pines de conexión.

Por su parte, se ha empleado la siguiente herramienta:

- Osciloscopio (el nuestro es digital).
- Fuente de tensión constante.
- Computador (tanto para alimentar, como para controlar a Arduino).
- Polímetro.
- Soldador.
- Estaño para soldar.

Los componentes los conectaremos mediante un cableado en una de las caras de la placa.

El esquema de montaje que he realizado se refleja en la siguiente imagen:

El montaje físico se muestra en las siguientes imágenes:

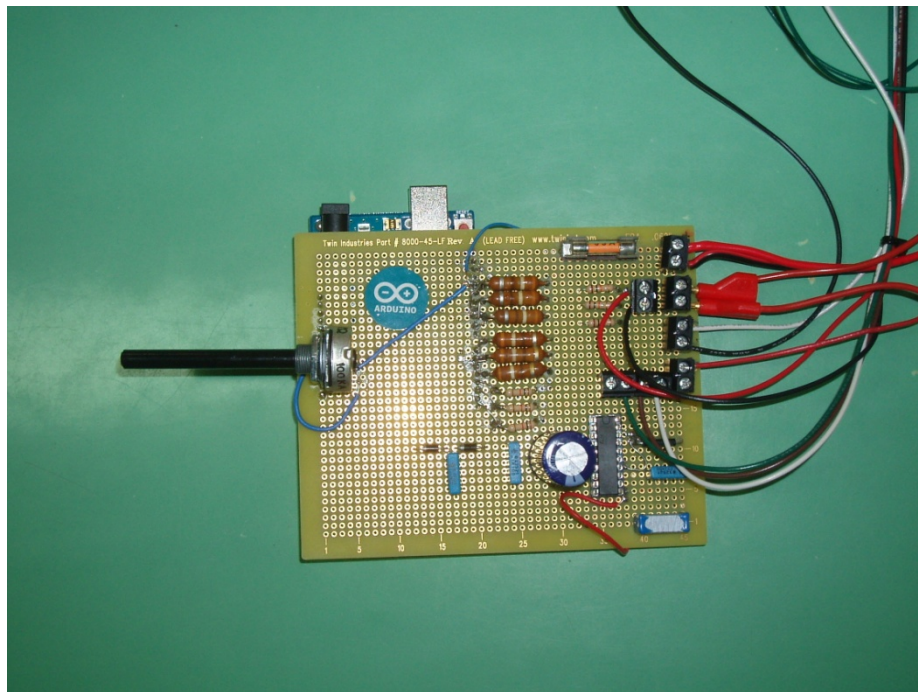


Fig. 3.3-Vista superior del controlador

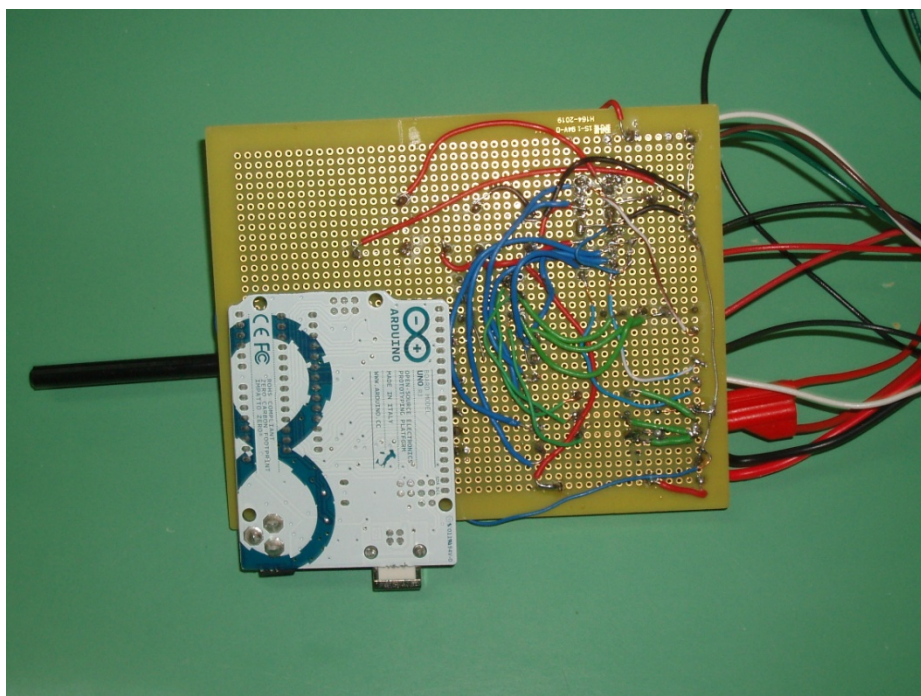


Fig. 3.4-Vista inferior del controlador

Para una mejor comprensión del funcionamiento del conjunto se explica por partes.

En primer lugar presentamos los bloques inversor y motor de manera simplificada para que resulte más fácil entender cómo trabajan conjuntamente:

El motor BLDC compuesto por tres fases (A, B, C) va tener activadas siempre dos de ellas. Se determinará que fases se deben activar mediante cada uno de los 6 vectores que se aplicaran al inversor activando o desactivando los correspondientes transistores. Los de nuestro inversor son de tipo DMOS

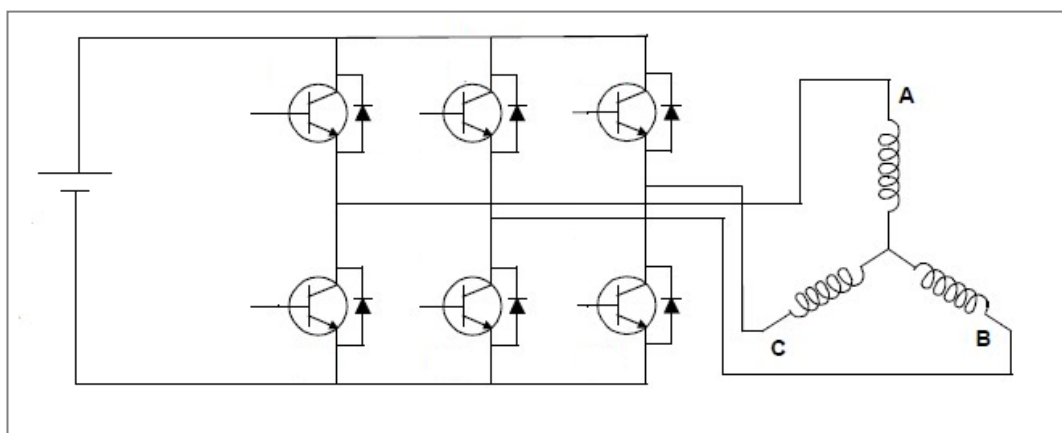


Fig. 3.5-Conexión entre el inversor y las bobinas del motor

Una pregunta que puede surgir al respecto es la siguiente:

-¿Cómo podemos saber qué vector aplicar en cada momento?

La respuesta es que, dependiendo de la posición del rotor respecto al estator, la cual nos la va a indicar los tres sensores Hall integrados en el motor; se aplicará un vector u otro. Se puede elaborar una tabla (ver tabla 3.1) indicando qué transistores hay que activar cada 60° de giro del motor y sus correspondientes fases excitadas, teniendo en cuenta que cada 360° se repite la secuencia.

VECTOR	Sensor Hall C	Sensor Hall B	Sensor Hall A	Fase A	Fase B	Fase C
4	1	0	0	-V	+V	NC
5	1	0	1	NC	+V	-V
1	0	0	1	+V	NC	-V
3	0	1	1	+V	-V	NC
2	0	1	0	NC	-V	+V
6	1	1	0	-V	NC	+V

Tabla 3.1- Relación vectores-activación fases (bobinas)

Como se observa cada 60° de giro de motor se aplica un vector. El número del vector lo marca los sensores Hall en binario, es decir, el valor de cada sensor es 0 o 1. Si los sensores nos marcan C=1, B=0, A=0, que conforma el número binario 100, en decimal se corresponde con el número 4; por tanto forma el vector 4.

Para mostrar más claramente cómo se comporta el conjunto, pongamos un ejemplo con uno de los vectores:

Imaginemos que los sensores nos marcan la posición del el vector 5, es decir el 101. Con este vector la fase A no habrá que inducir la tensión. Sin embargo la fase B tiene una excitación con tensión positiva, y la fase C negativa.

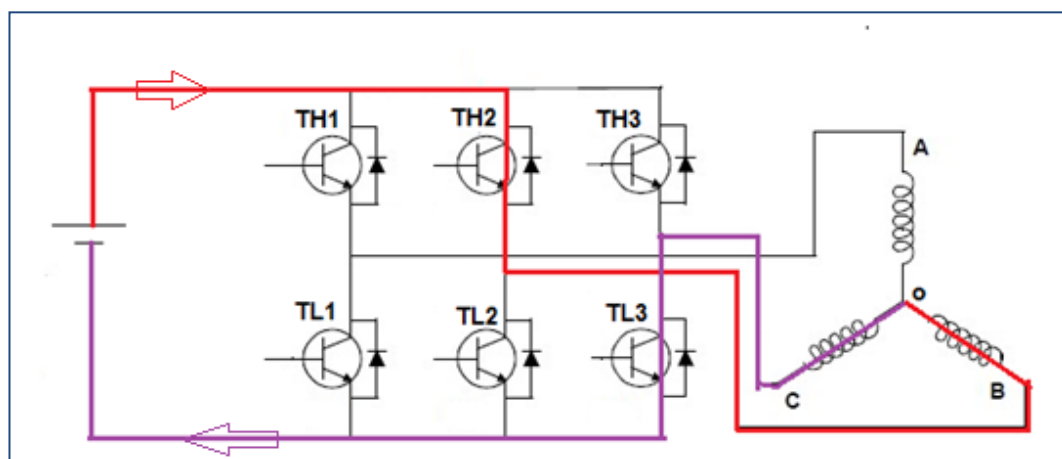


Fig. 3.6-Corrientes al aplicar vector 101

Marcamos en la figura 3.6 de color rojo el recorrido de la tensión positiva y de violeta el de la tensión negativa, siempre y cuando tomemos como referencia el punto o.

Analizando la figura 3.6 se entiende que se deben activar los transistores TH2 y TL3 (del inversor). El encargado de activarlos será Arduino.

Para determinar cómo activa Arduino estos transistores, tomamos el esquema del inversor real.

Hemos mantenido la nomenclatura **TH-TL** de los transistores para evitar confusiones entre el modelo y el inversor real. Además esta forma de nombrarlos hace que sea más intuitiva su interpretación. Los transistores TH- Transistor High son los que se encuentran situados en la parte superior. Para activarlos es preciso, a través de Arduino, introducir un 1 lógico, es decir un nivel alto (High level). Los transistores TL- Transistor Low, se activan mediante un 0 lógico (Low level) con Arduino.

Las entradas ENable del inversor activan cada rama que contiene su TH-TL. Con la entrada IN del inversor se activa cada uno de los seis transistores.

Continuando con la explicación del vector 5, hemos indicado que los transistores TH2 y TL3 deben ser activados. Recordando los pasos a seguir, ya explicados en el capítulo de Arduino. Lo primero que hay que hacer es habilitar los de transistores TH2 TL2, TH3 Y TL3 mediante su correspondiente entrada *ENable*, es decir, EN 2 y EN3 con un 1 lógico. A continuación hay que elegir que transistor del bloque 2 y 3 queremos que se active con la entrada *INput*. En este caso son TH2, por lo que necesita $IN2=1$; y TL3 mediante $IN3=0$.

Al no requerir la fase A; $EN1=0$ para deshabilitar los transistores TH1 y TL1, de forma que $IN1=X$, es decir, que no importa el valor que pongamos en $IN1$.

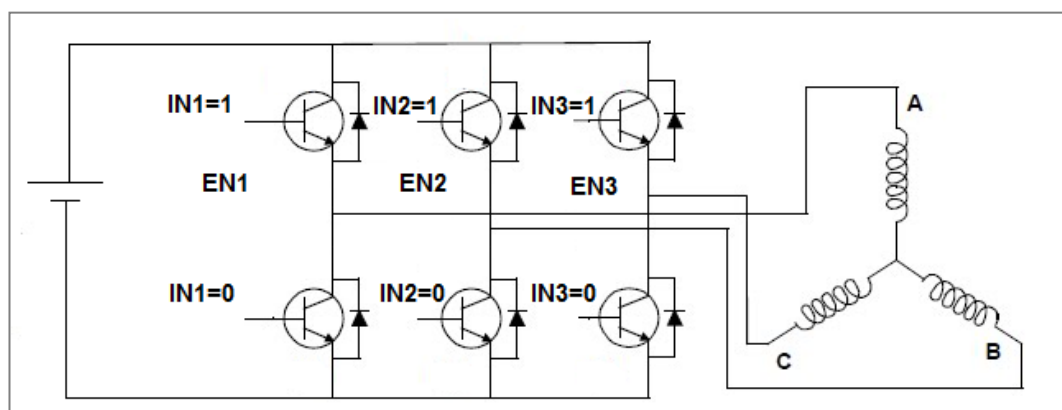


Fig. 3.7-Nomenclatura de las ramas y transistores del inversor.

En la figura 3.8 se muestra el inversor real, en el que la fase A corresponde con la salida del inversor OUT 1, la fase B con OUT 2 y la fase C con el OUT 3.

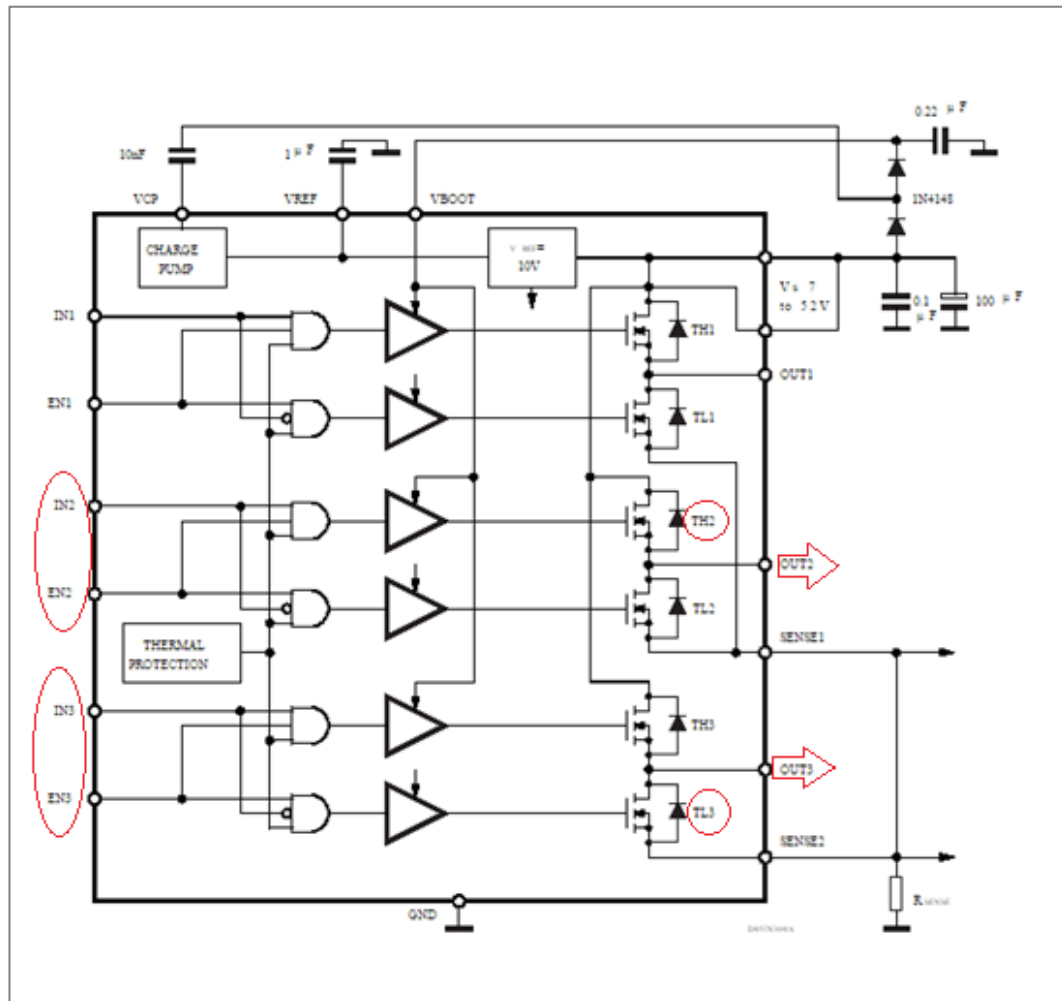


Fig. 3.8-Nomenclatura del inversor real

Este sería el procedimiento a seguir, para controlar el giro del motor. Pero el propósito de este proyecto, no sólo es que gire el motor, sino además controlar su velocidad. Este control lo vamos a realizar variando el ciclo de servicio de la función PWM de Arduino. Recordamos que la función PWM consiste en generar una onda cuadrada de forma que simule una señal analógica con una resolución de 28 bits, es decir, posee 256 posibles valores. De esta manera si el PWM=0, equivale a 0 voltios (0% del ciclo de servicio); y PWM= 255 equivale a 5 voltios (100% del ciclo de servicio).

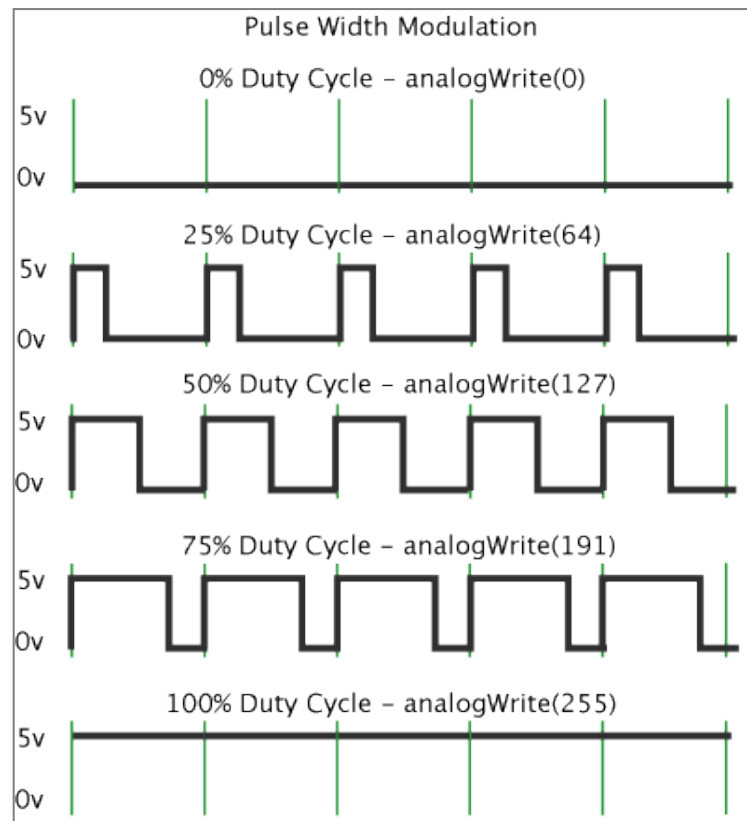


Fig. 3.9-Recordatorio del concepto PWM

Físicamente, la velocidad la controlaremos con un potenciómetro, que simulará el acelerador de una moto eléctrica. Arduino, mediante un mapeo del potenciómetro, sabrá que velocidad estamos pidiendo en cada momento.

Una vez que había finalizado el montaje del circuito, necesitaba comprobar si realizaba las funciones correctamente. Para ello en vez de conectar al circuito el motor, le conecté tres resistencias simulando las fases del motor. Alimentando el circuito a 12 voltios, introduje el siguiente fragmento de programa para imitar el vector 5 que acabamos de explicar:



```
void setup(){  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(10, OUTPUT);  
  pinMode(11, OUTPUT);  
}  
  
void loop(){  
  
  digitalWrite(7, LOW);  
  digitalWrite(6, HIGH);  
  digitalWrite(5, LOW);  
  
  analogWrite(11,255);  
  analogWrite(10,127);  
  analogWrite(9,0);  
}
```

Donde los pines 5, 6 y 7 corresponden a IN1, IN2 e IN3 respectivamente. Por otro lado los pines 9, 10, 11 son EN1, EN2 y EN3.

Analizando el fragmento de programa, destacamos que hacemos el PWM sobre los EN. Variaremos el ciclo de servicio, sobre la pareja de transistores TH-TL (EN1, EN2, EN3) de una de las fases excitadas, en la cual el transistor esté a nivel alto, es decir, activado con su correspondiente IN=1. En aquella pareja encargada de excitar la otra fases, cuyo IN=0, el PWM se hará al 100%.

Tomamos las siguientes medidas con el osciloscopio. La primera imagen corresponde a la fase B (amarillo) y la fase C (azul). Tomando como referencia el punto común a las fases corroboramos que funciona correctamente, puesto que C tiene la misma amplitud y valor de tensión, pero de signo contrario.

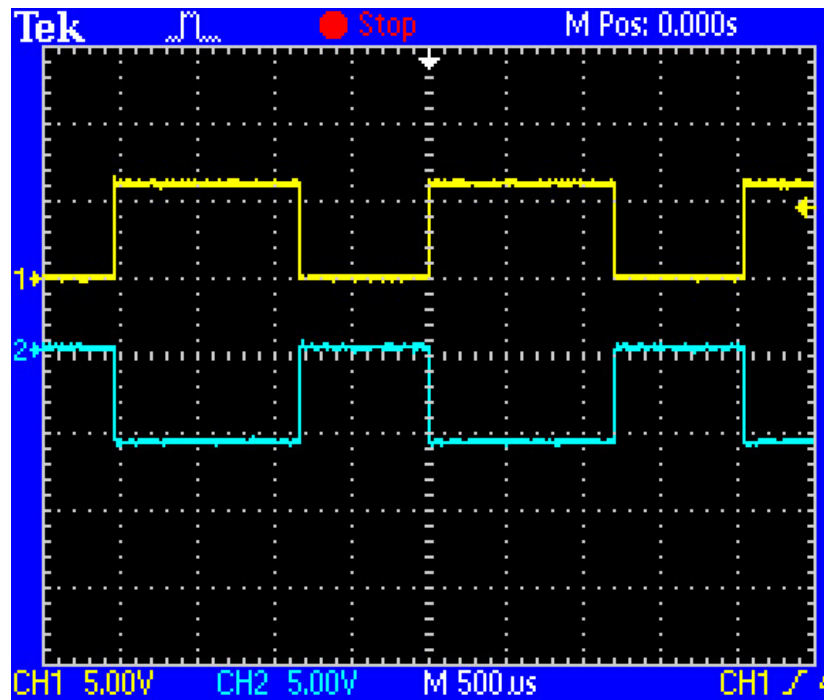


Fig. 3.10-Forma de onda las fases A-B del motor obtenidas con el osciloscopio

Por otro lado tomamos medidas de fase B (amarillo), y fase A (azul). Efectivamente la fase A se mantiene a 0, puesto que no le corresponde a este vector su excitación.

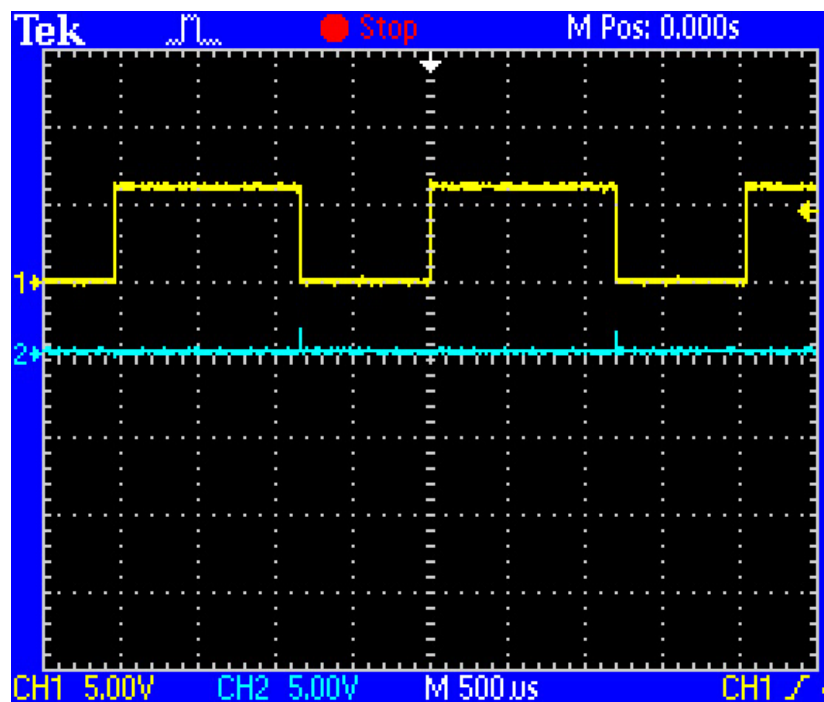


Fig. 3.11-Forma de onda de las fases B-C del motor obtenidas con el osciloscopio



Las medias sólo las hemos podido tomar de dos en dos, ya que el osciloscopio que manejamos solamente posee dos sondas.

Para obtener el giro completo y continuo del motor, pudiendo variar la velocidad se crea el siguiente código en Arduino:

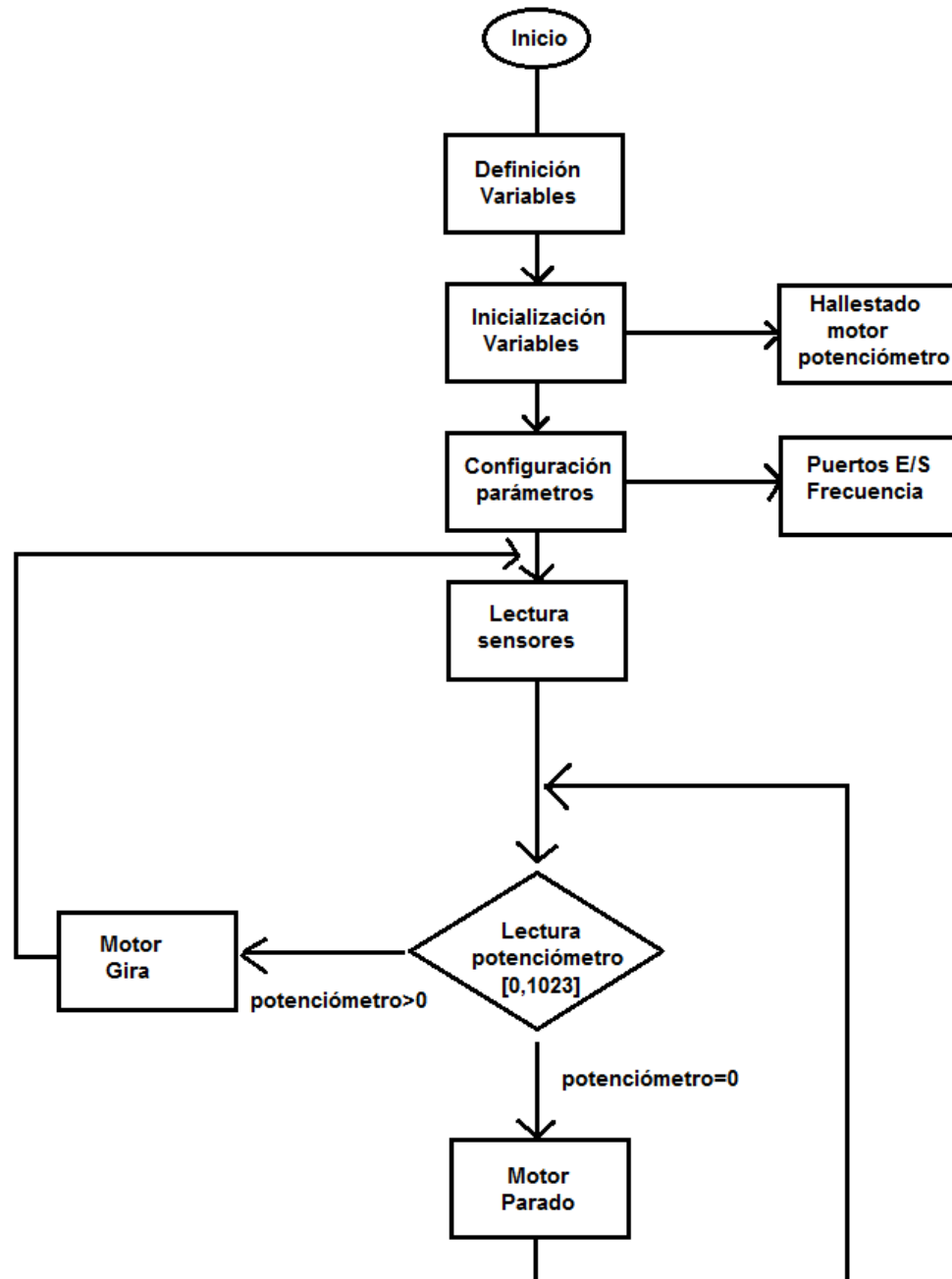


Fig. 3.12-Diagrama de bloques del programa principal



/*

Programa para el control de motor BLDC

TFG Jesus Carlos Rodríguez Pintor

Grado en Ingeniería Electrónica Industrial Y Automática

Universidad de Valladolid

Febrero 2015

*****/

//declaracion de variables

int Halleestado1; //sensor1

int Halleestado2; //sensor2

int Halleestado3; //sensor3

int Halleestado = 1; //estado binario de los tres sensores, inicializamos a 1

int motor = 0; //velocidad del motor, inicializamos a 0

int acelerador = 0; //variable que representa el potenciómetro

void setup() {

//entradas

pinMode(1,INPUT); // Hall 1

pinMode(2,INPUT); // Hall 2

pinMode(3,INPUT); // Hall 3



```
// salidas al L6234
```

```
pinMode(5,OUTPUT); // IN 1
```

```
pinMode(6,OUTPUT); // IN 2
```

```
pinMode(7,OUTPUT); // IN 3
```

```
pinMode(9,OUTPUT); // EN 1
```

```
pinMode(10,OUTPUT); // EN 2
```

```
pinMode(11,OUTPUT); // EN 3
```

```
int prescalerVal = 0x07; //variable con numero binario "00000111"
```

```
TCCR1B &= ~prescalerVal; //funcion AND entre prescaler y el TIMER1 11111000"
```

```
int prescalerVal2 = 2; //variable que le asignamos el numero "00000001"
```

```
TCCR1B |= prescalerVal2; //funcion OR entre prescaler y el TIMER1 "00000001"
```

```
//Hacemos lo mismo con el TIMER2
```

```
TCCR2B &= ~prescalerVal;
```

```
TCCR2B |= prescalerVal2;
```

```
}
```

```
void loop(){
```

```
acelerador = analogRead(0); //lectura del potenciómetro
```

```
motor = map(acelerador, 0, 1023, 0, 255);
```

```
//Lectura de los 3 sensores
```

```
Hallestado1 = digitalRead(1);
```

```
Hallestado2 = digitalRead(2);
```

```
Hallestado3 = digitalRead(3);
```




//Creamos la variable que conforma los tres estados:

*Hallestado = (Hallestado1) + (2*Hallestado2) + (4*Hallestado3);*

//según la posición del rotor indicada por los sensores

//introducimos el vector correspondiente

switch (Hallestado) {

case 5:

//vector 5

digitalWrite (5,LOW);

digitalWrite (6,HIGH);

digitalWrite (7,LOW);

analogWrite (9,0);

analogWrite (10,motor);

analogWrite (11,255);

break;

case 4:

//vector 4

digitalWrite (5,LOW);

digitalWrite (6,HIGH);

digitalWrite (7,LOW);

analogWrite (9,255);

analogWrite (10,motor);



```
analogWrite (11,0);
```

```
break;
```

```
case 6:
```

```
//vector 6
```

```
digitalWrite (5,LOW);
```

```
digitalWrite (6,LOW);
```

```
digitalWrite (7,HIGH);
```

```
analogWrite (9,255);
```

```
analogWrite (10,0);
```

```
analogWrite (11,motor);
```

```
break;
```

```
case 3:
```

```
//vector 3
```

```
digitalWrite (5,HIGH);
```

```
digitalWrite (6,LOW);
```

```
digitalWrite (7,LOW);
```

```
analogWrite (9,motor);
```

```
analogWrite (10,255);
```

```
analogWrite (11,0);
```

```
break;
```



case 2:

//vector 2

digitalWrite (5,LOW);

digitalWrite (6,LOW);

digitalWrite (7,HIGH);

analogWrite (9,0);

analogWrite (10,255);

analogWrite (11,motor);

break;

case 1:

//vector 1

digitalWrite (5,HIGH);

digitalWrite (6,LOW);

digitalWrite (7,LOW);

analogWrite (9,motor);

analogWrite (10,0);

analogWrite (11,255);

break;

}

}

Una vez que introducimos este código en arduino, y alimentamos tanto la placa como el motor de la forma que se muestra en la figura 3.13, comprobamos su funcionamiento. La alimentación de los sensores será a 5V, y la de la placa y el motor a 12 V. Arduino se alimenta desde el USB que le conecta al PC

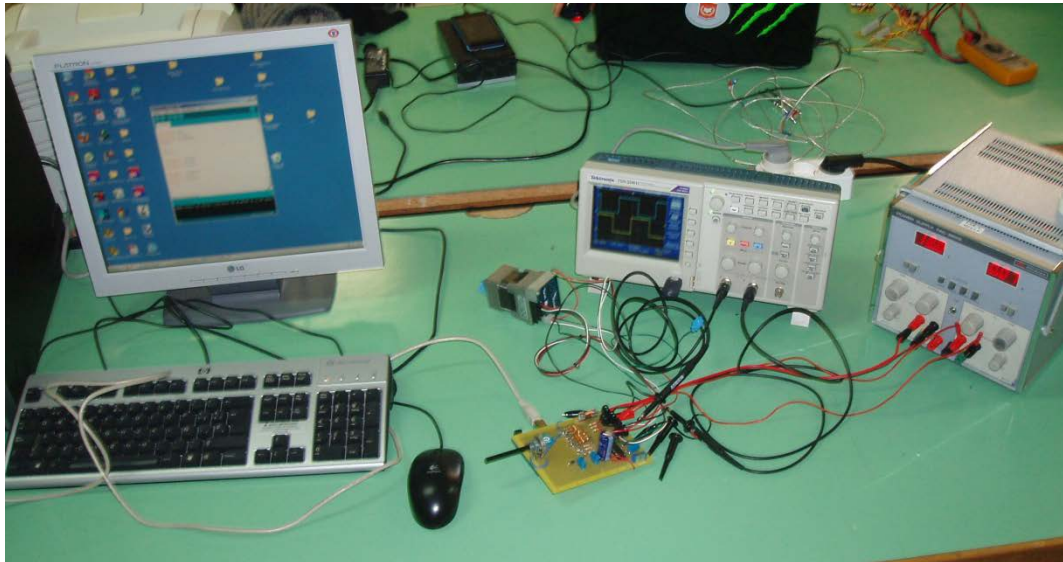


Fig. 3.13-Vista del conjunto que conforma el controlador del motor, el motor y la alimentación

En la imagen 3.14 se observa la alimentación a 12v. Hay que tener en cuenta que las sondas son de tipo x10, es decir, multiplican la señal que observan por 10.

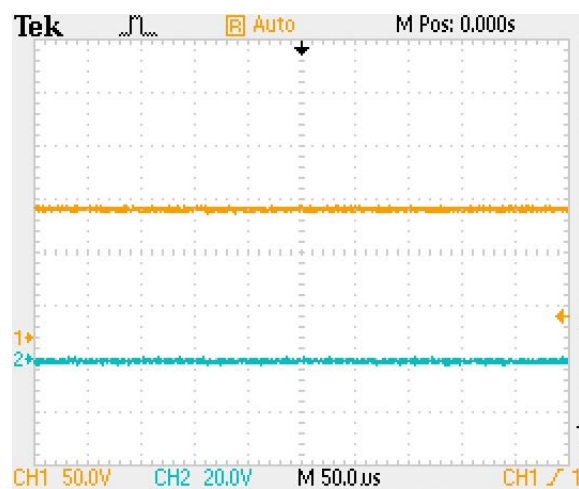


Fig. 3.14-nivel de tensión a la entrada de la placa

La forma cuadrada de los sensores Hall se comprueba en la figura 3.15 y 3.16. La diferencia entre ambas figuras es la diferencia de velocidad en el giro del motor, es por ello que en la 3.14 la frecuencia es mayor, es decir el periodo de la señal es inferior:

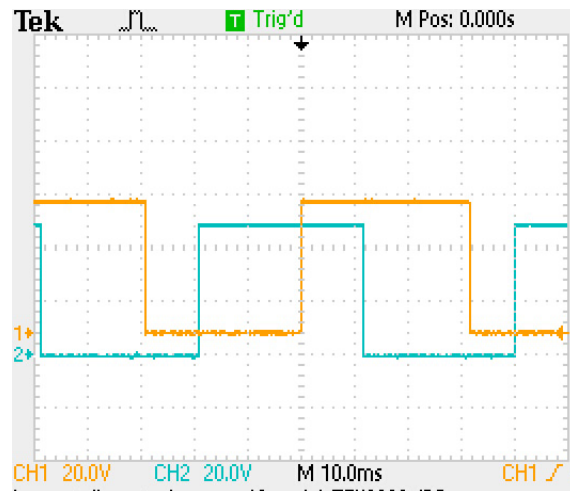


Fig. 3.15-forma de onda de dos sensores a bajas revoluciones del motor

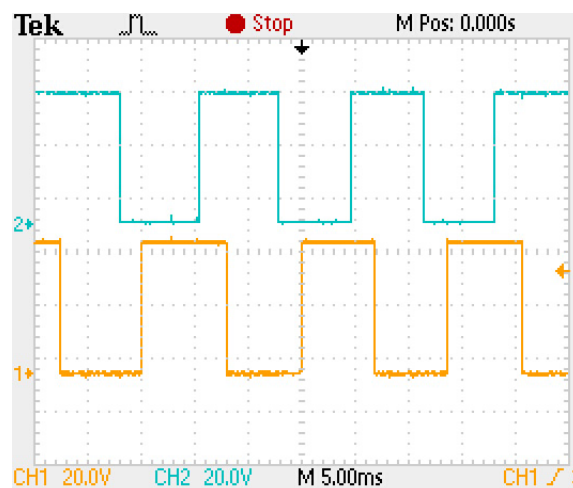


Fig. 3.16-forma de onda de los sensores a altas revoluciones del motor

Como indicamos anteriormente, el PWM lo llevaremos a cabo en las entradas EN del inversor. Tomamos una imagen de lo que ocurre en el pin EN1 en un determinado instante



Fig. 3.17-forma de onda del PWM en EN1

Para concluir mostraremos los impulsos en las entradas IN1 e IN2 del inversor generados por Arduino, véase la figura 3.18.

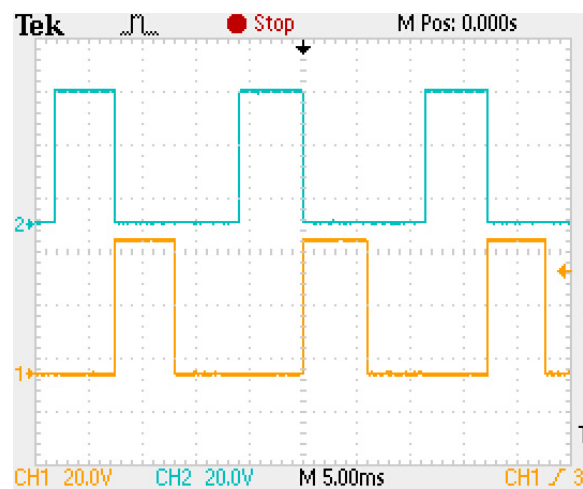


Fig. 3.18-forma de onda de los impulsos aplicados a IN1 e IN2

El último paso con el que concluimos el proyecto ha sido conformar un programa con el que conseguimos introducirnos al **frenado regenerativo**.

El frenado regenerativo consiste en devolver energía a la batería del vehículo cuando éste decelera, convierte la energía cinética nuevamente en eléctrica; es decir, se convierte en generador. Puesto que no he utilizado una batería para alimentar el sistema, sino que he usado una fuente de tensión alimentada a red; no la puedo devolver energía a esta fuente ya que probablemente quedara inutilizable. También sería necesario hacer girar el eje del motor suficientemente rápido como para generar electricidad en las bobinas del motor. Esta velocidad no se puede alcanzar girando manualmente el eje del motor, sino que habría que acoplar otro motor DC para alcanzar las revoluciones deseadas. Es por estos aspectos técnicos que no podré mostrar el funcionamiento del frenado regenerativo.

Pese a ello se presenta un código que pretende poder realizar este tipo de freno. Esta parte del TFG, puesto que es complementaria, simplemente se inicia para una futura continuación en el tema.

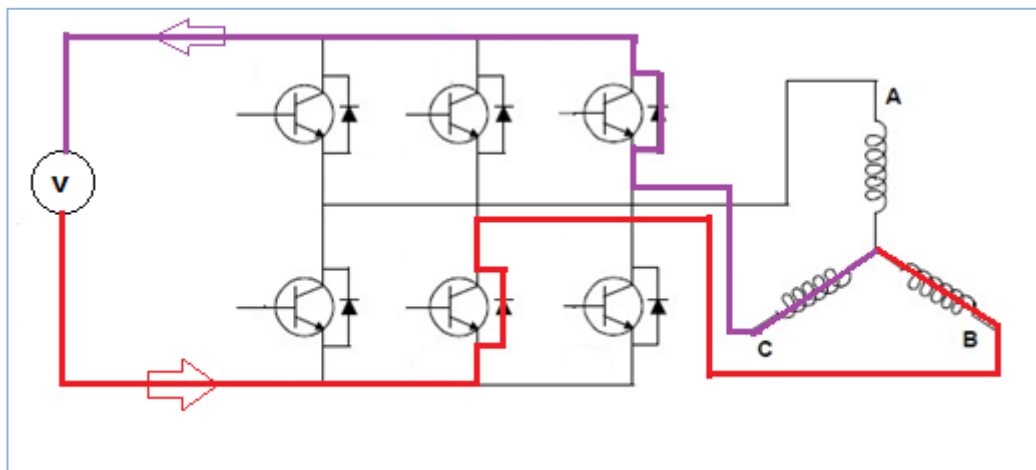


Fig. 3.19- Sentido de circulación de la corriente en el frenado regenerativo

Fijándonos en el esquema (fig 3.19), se aprecia como el sentido de la corriente es en sentido contrario funcionando como generador, al sentido de cuando funciona como motor. Además la corriente circulará por los diodos en vez de por los transistores

El frenado regenerativo puede resultar muy útil en conducción por ciudad, puesto que puede llegar a suponer recuperar hasta el 30% de energía

cinética. Además este tipo de frenos pesa un 22% menos que un freno convencional de los que utilizan los vehículos actuales.

Para poder realizar el frenado regenerativo, debemos modificar ligeramente el programa. Añadiremos un bucle “if” para elegir, dependiendo de la posición del potenciómetro si queremos que la máquina eléctrica funcione como motor o como generador. El programa final sería el siguiente:

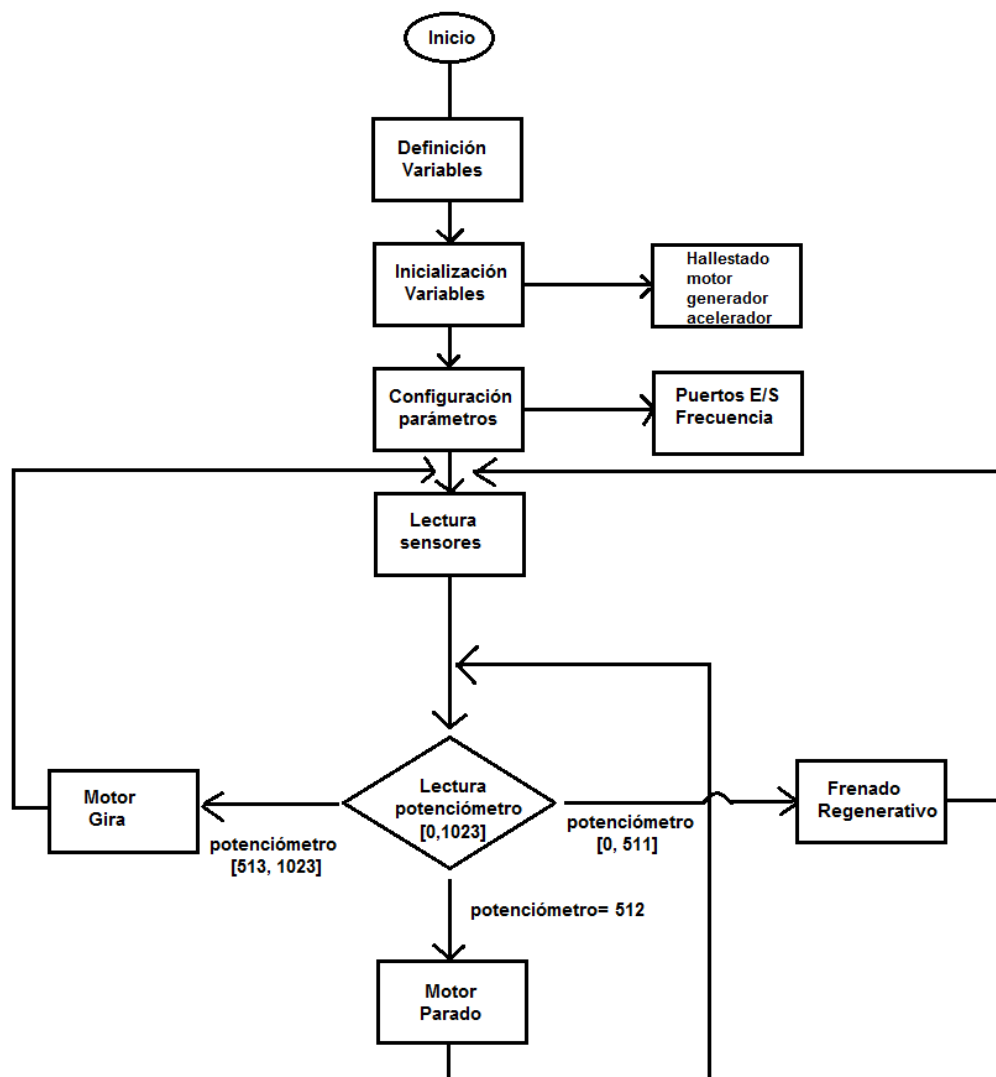


Fig. 3.20- Diagrama de bloques programa con frenado regenerativo



/******

PROGRAMA DE CONTROL DE MOTOR BLDC CON FRENADO REGENERATIVO

TFG JESUS CARLOS RODRIGUEZ PINTOR

GRADO EN INGENIERÍA ELECTRONICA INDUSTRIAL Y AUTOMATICA

UNIVERSIDAD DE VALLADOLID

FEBRERO 2015

*****/

```
int Hallestado1;

int Hallestado2;

int Hallestado3;

int Hallestado = 1;

int motor = 0;

int generador=0;

int acelerador = 0; //es el potenciómetro

void setup() {

    pinMode(1,INPUT); // Hall 1

    pinMode(2,INPUT); // Hall 2

    pinMode(3,INPUT); // Hall 3

    pinMode(5,OUTPUT); // IN 1

    pinMode(6,OUTPUT); // IN 2

    pinMode(7,OUTPUT); // IN 3
```



```
pinMode(9,OUTPUT); // EN 1  
  
pinMode(10,OUTPUT); // EN 2  
  
pinMode(11,OUTPUT); // EN 3
```

```
int prescalerVal = 0x07;  
  
TCCR1B &= ~prescalerVal;  
  
int prescalerVal2 = 2;  
  
TCCR1B |= prescalerVal2;  
  
TCCR2B &= ~prescalerVal;  
  
TCCR2B |= prescalerVal2;  
  
}
```

```
void loop(){  
  
    acelerador = analogRead(0);  
  
    motor = map(acelerador, 513, 1023, 0, 255);  
  
    generador = map(acelerador, 0, 511,255,0);  
  
  
    Hallestado1 = digitalRead(1);  
  
    Hallestado2 = digitalRead(2);  
  
    Hallestado3 = digitalRead(3);  
  
  
    Hallestado = (Hallestado1) + (2*Hallestado2) + (4*Hallestado3);  
  
    //dependiendo de la posición del potenciómetro es motor o generador  
  
    if(acelerador > 511){
```



```
//actúa como motor

switch (Hallestado)

{

case 5:

//vector 5

digitalWrite (5,LOW);

digitalWrite (6,HIGH);

digitalWrite (7,LOW);


analogWrite (9,0);

analogWrite (10,motor);

analogWrite (11,255);

break;


case 4:

//vector 4

digitalWrite (5,LOW);

digitalWrite (6,HIGH);

digitalWrite (7,LOW);


analogWrite (9,255);

analogWrite (10,motor);

analogWrite (11,0);
```



break;

case 6:

//vector 6

digitalWrite (5,LOW);

digitalWrite (6,LOW);

digitalWrite (7,HIGH);

analogWrite (9,255);

analogWrite (10,0);

analogWrite (11,motor);

break;

case 3:

//vector 3

digitalWrite (5,HIGH);

digitalWrite (6,LOW);

digitalWrite (7,LOW);

analogWrite (9,motor);

analogWrite (10,255);

analogWrite (11,0);

break;



case 2:

//vector 2

digitalWrite (5,LOW);

digitalWrite (6,LOW);

digitalWrite (7,HIGH);

analogWrite (9,0);

analogWrite (10,255);

analogWrite (11,mSpeed);

break;

case 1:

//vector 1

digitalWrite (5,HIGH);

digitalWrite (6,LOW);

digitalWrite (7,LOW);

analogWrite (9,motor);

analogWrite (10,0);

analogWrite (11,255);

break;

}

}

else{



```
//actúa como generador

switch (Hallestado)

{

    case 5:

        //vector 5

        analogWrite (9,0);

        analogWrite (10,generador);

        analogWrite (11,0);

        break;


    case 4:

        //vector 4

        analogWrite (9,0);

        analogWrite (10,generador);

        analogWrite (11,0);

        break;


    case 6:

        //vector 6

        analogWrite (9,0);

        analogWrite (10,0);

        analogWrite (11,generador);

        break;
```



case 3:

//vector 3

analogWrite (9,generador);

analogWrite (10,0);

analogWrite (11,0);

break;

case 2:

analogWrite (9,0);

analogWrite (10,0);

analogWrite (11,generador);

break;

case 1:

//vector 1

analogWrite (9,generador);

analogWrite (10,0);

analogWrite (11,0);

break;

}

}

}



[PÁGINA INTENCIONADAMENTE EN BLANCO]



4

SIMULACIÓN CON MATLAB

Entendemos por simulación, la experimentación dentro de un sistema a través de un programa de simulación.

Con la herramienta *simulink* de matlab vamos a simular el comportamiento del sistema electrónica que hemos ido describiendo a lo largo del TFG.

La simulación me ha parecido importante ya que me permite manipular el sistema sin riesgo de provocar averías.

Por ejemplo, puedo variar tensiones o corrientes sin el peligro de provocar cortocircuitos o sobretensiones.

También me muestra la evolución del sistema ante diferentes situaciones o la sensibilidad de los parámetros que lo componen.

Además cuento con una serie de instrumentos de medida que me facilita, tanto a través de datos numéricos como de gráficas, la comprensión en conjunto del sistema.

Otro factor que hace de la simulación una herramienta indispensable, es que puedo aislar componentes y manipular sus variables de manera independiente.

Por último, una posibilidad que propicia trabajar empleando la simulación es la de modificar la escala de tiempo permitiendo apreciar matices que en escala real no se podrían observar. Es el ejemplo de comprobar, por ejemplo, el desgaste del motor en años o examinar los impulsos de PWM a frecuencia de 32kHz cuya velocidad es inapreciable por el ojo humano.

El único inconveniente que podría citar de la simulación es la dificultad de introducir en el modelo la gran variedad de adversidades inesperadas que podrían surgir durante la ejecución real del sistema y a la que habrá que afrontar.

Volviendo a nuestro sistema simulado, haremos un estudio de cada componente que lo conforma y veremos su paralelismo con el sistema físico



real. A continuación mostraremos todos los bloques en conjunto y correctamente unidos. Con las gráficas analizaremos la similitud al real.

En primer lugar tenemos la **batería** de tensión continua. Podemos variar antes de comenzar la simulación el nivel de voltaje que suministra esta fuente.

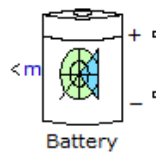


Fig. 4.1- Representación batería Simulink

El siguiente componente es un **puente o inversor universal**. Tiene la característica de poder elegir el tipo de componente para conformar este convertidor DC/AC, como MOSFET, IGBT, diodos...etc. El posible además elegir el número de ramas, que en nuestro caso serán tres puesto que es trifásico.

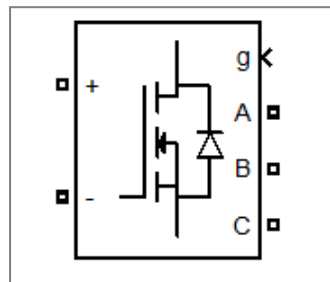


Fig. 4.2- Representación inversor simulink

El elemento que se muestra a continuación es el **motor**. Lo que está conectado a él, es un decodificador; el cual lo utilizaremos para poder visualizar diferentes variables al conectar un elemento creador de gráficas llamado *scope*. Es posible elegir entre el *par* o la *velocidad*, como la variable de entrada al motor. Se puede comprobar que posee las tres fases que conforman motor BLDC. Se le puede inducir una par en la señal T_m mediante una señal *step*.

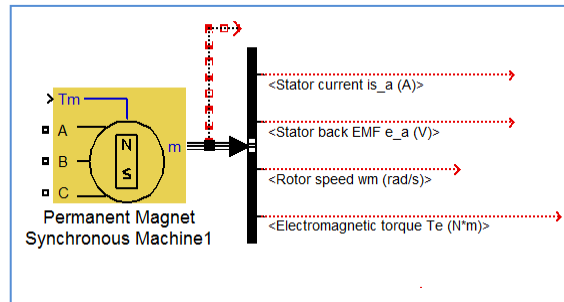


Fig. 4.3- Representación motor simulink

El bloque **decoder** tiene como entrada las señales provenientes de los sensores Hall, que denominaremos ha, hb, hc. Como salida las señales de activación de cada fase del motor, en este caso las denominaremos emf_a, emf_b, emf_c; donde “emf” es la simplificación de “fuerza electromotriz”.

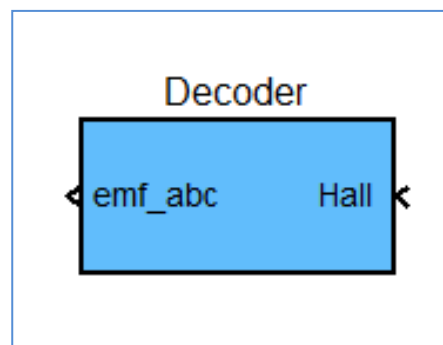


Fig. 4.4- Representación bloque decoder simulink

Al igual que hicimos con el modelo real, dependiendo de la posición que marquen los sensores, deberemos activar la bobina que cree el mayor par posible. Se adjunta la tabla 4.1 prediseñada por el simulador con la activación de las bobinas del motor.

ha	hb	hc	emf_a	emf_b	emf_c
0	0	0	0	0	0
0	0	1	0	-1	+1
0	1	0	-1	+1	0
0	1	1	-1	0	+1
1	0	0	+1	0	-1
1	0	1	+1	-1	0
1	1	0	0	+1	-1
1	1	1	0	0	0

Tabla 4.1-Tabla de vectores con su correspondiente activación de bobinas

La configuración interna del bloque se muestra con este aspecto:

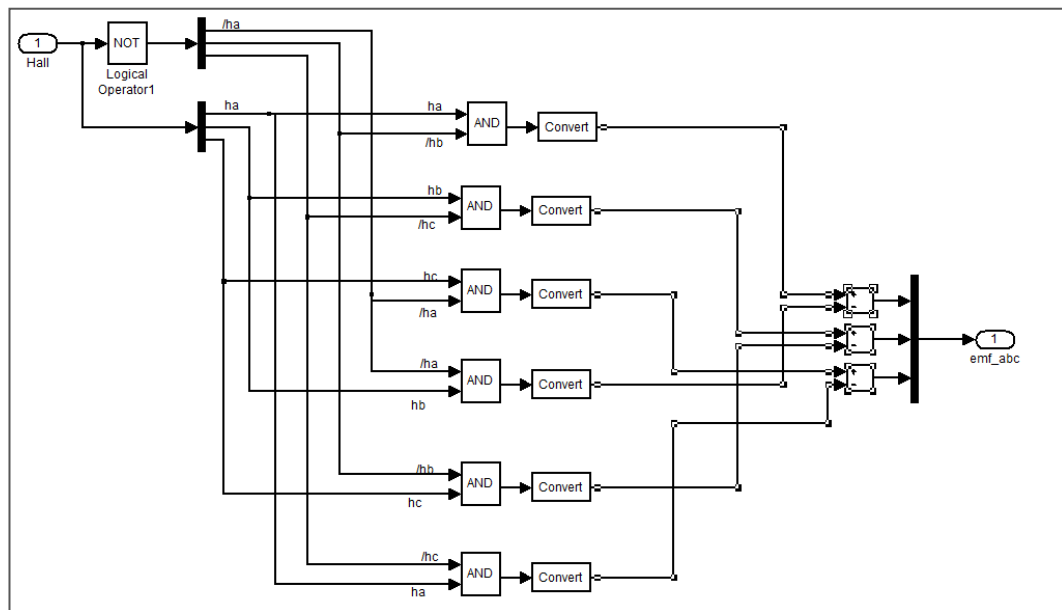


Fig. 4.5- Configuración interna bloque decoder

El esquema se modela de forma que queramos activar una u otra bobina según la señal de los sensores.

Como su propio nombre indica, *decoder* se encarga de decodificar la señal de los sensores a las bobinas del motor.

Ya indicamos que el motor posee tres bobinas. Cada una de ellas se compone de dos polos. De esos dos polos a uno se le puede alimentar con tensión positiva o negativa (+1, -1), dependiendo de la posición del imán de rotor. Para indicar sobre que polo hay introducir una señal, se necesita el bloque **gates**.

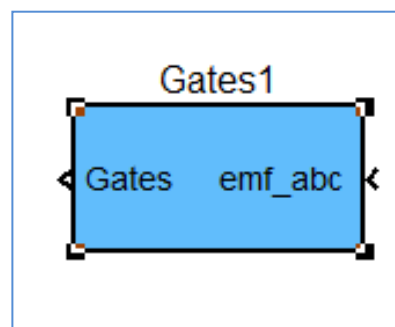


Fig. 4.6- Representación bloque gates simulink

Este bloque se coloca a continuación del bloque *decoder*. Es el que se encargará de traducir la señal de la bobina (+1, -1) para cada polo (0, 1). La tabla que aparece intrínseca a este bloque, y del cual la he extraído es la siguiente:

emf_a	emf_b	emf_c	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	0	0	0
0	-1	+1	0	0	0	1	1	0
-1	+1	0	0	1	1	0	0	0
-1	0	+1	0	1	0	0	1	0
+1	0	-1	1	0	0	0	0	1
+1	-1	0	1	0	0	1	0	0
0	+1	-1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0

Tabla 4.2-Correspondencia entre activación de bobinas y polos de potencia

Los polos Q1 y Q2 pertenecen la bobina denominada “a”, Q3 y Q4 pertenecen a la bobina “b”; y por último Q5 y Q6 a la bobina “c”.

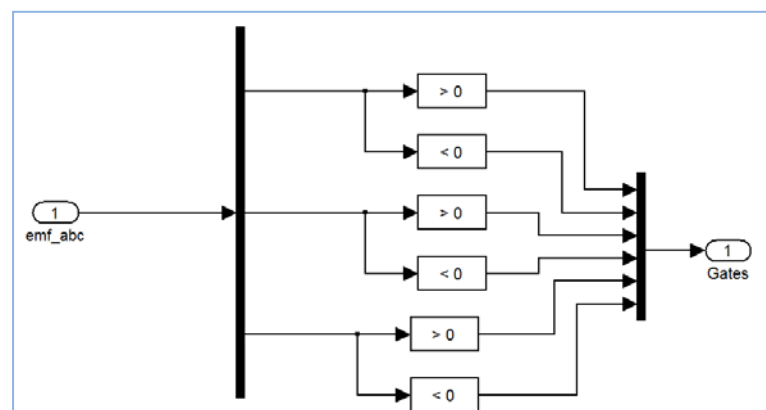


Fig. 4.7- Configuración interna del bloque gates

Dependiendo de si la señal emf es +1 o -1, implica a que polo pertenece, y en este caso +1 indica los polos superiores y -1 los inferiores.

Una vez conocido las señales o impulsos para aplicar, necesitamos conocer la **tensión de control** para poder realizar un PWM correctamente. La *tensión de control* se obtiene mediante un conjunto conformado por una serie de bloques:

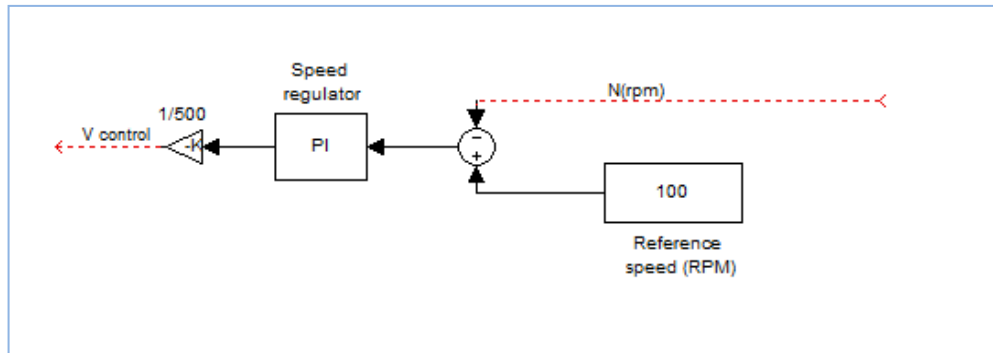


Fig. 4.8- Bucle realimentación PI

Hay una referencia de velocidad, que mediante un comparador se confronta con la que tiene el motor en cada momento. El resultado de la comparación pasa al bloque PI (proporcional-integral) y después pasa por un bloque de ganancia.

Como el modo de control para el motor se basa en un PWM, el único bloque que os falta por colocar para completar el modelo de la simulación es el de **PWM control**.

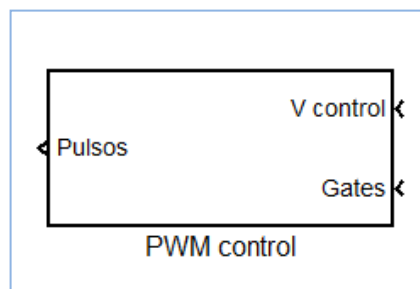


Fig. 4.9- Representación bloque PWM simulink

Este elemento enviará los pulsos al inversor indicados por el bloque *gates*, a una tensión que marcará la *tensión de control* que acabamos de explicar.

El esquema interno del bloque se estructura como muestra la figura 4.10.

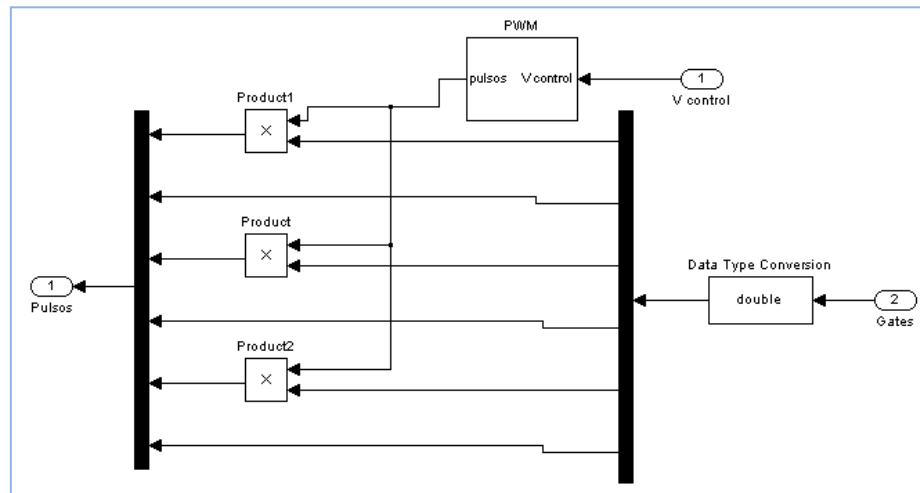
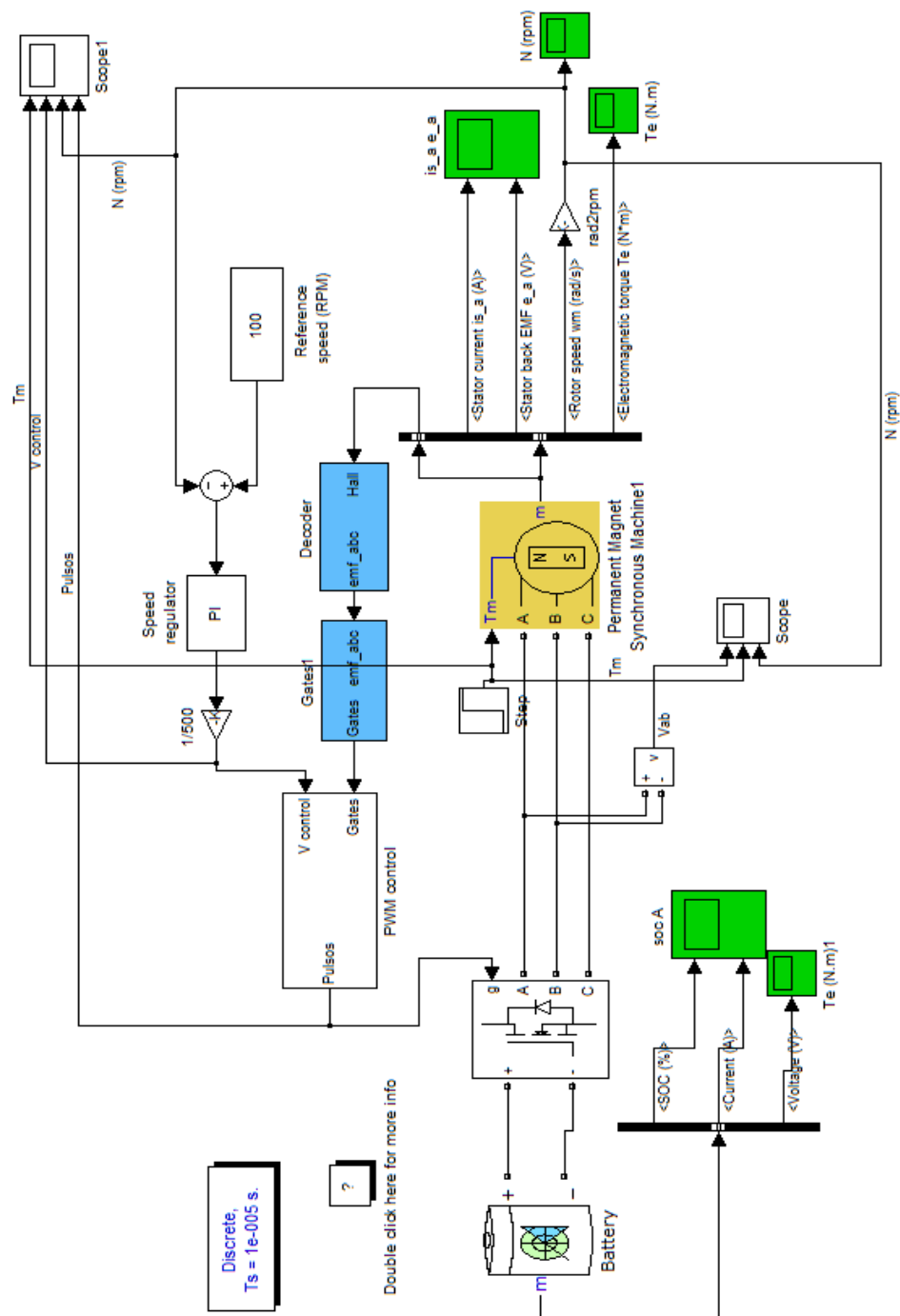


Fig. 4.10- Configuración interna bloque PWM

El PWM lo haremos sobre los polos superiores, al igual que en el modelo real. Por ello los polos inferiores no se multiplican por la tensión de control.

Las señales que provienen de *gates* son de tipo boolean, por lo que para poderlas multiplicar se necesita transformarlas a double.



Puesto que en la simulación podemos imponer los valores que queramos, sin correr el riesgo de destruir los componentes, introducimos en la batería una tensión de 500v.

Las gráficas obtenidas a partir de un scope son las siguientes:

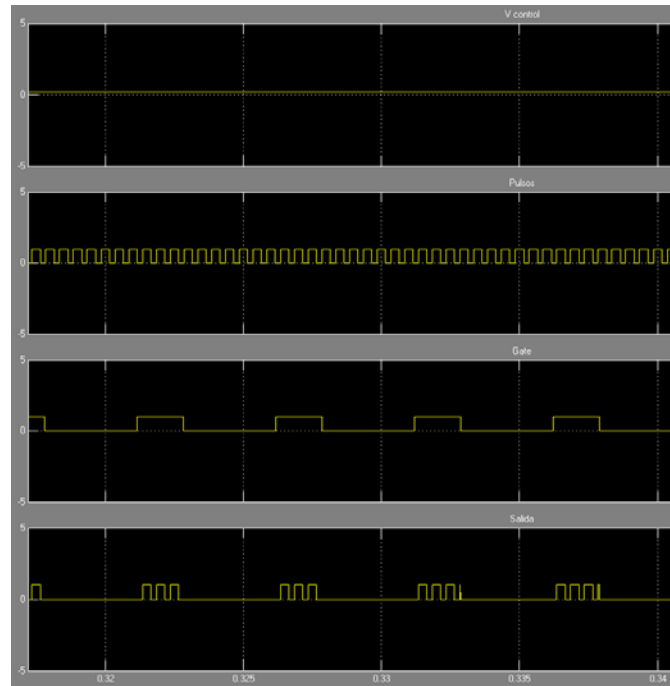


Fig. 4.12- Vcontrol, pulsos, Gate y Salida

Los pulsos equivaldrían a la función IN de arduino, y el PWM sería el realizado por los pines EN.



[PÁGINA INTENCIONADAMENTE EN BLANCO]



5

CONCLUSIONES

Hemos ido viendo las diferentes partes que constituyen un sistema electrónico que permite controlar la velocidad de un motor BLDC, para posteriormente comprender como interactúan entre ellas.

En primer lugar se desarrolla un modelo virtual, basado en el entorno Matlab; con la herramienta Simulink. Nos servirá como apoyo al modelo real.

El controlador construido físicamente se compone de una parte hardware y de otra software.

Este controlador de velocidad diseñado es robusto y versátil en dos sentidos:

Por un lado el **software** que hemos utilizado basado en Arduino, nos ofrece una programación que permite conectarse con infinidad de dispositivos a la vez, sin que se altere su funcionamiento. Además posee un lenguaje de código similar al C, es decir, es bastante intuitivo y no requiere de excesivos conocimientos previos para programar en este lenguaje.

Por otra parte, el **hardware** (se debe tener en cuenta que es de propia construcción y, por lo tanto, no está perfeccionado al máximo), nos ofrece una plataforma para poder indagar en otros proyectos.

Uno de estos proyectos podría ser indagar en cómo controlar motores *sensorless* (*sin sensores*). Como este tipo de motores no lleva sensores incorporados, habría que modificar el código, pero se podría mantener el material utilizado.

También se podría continuar con el frenado regenerativo. No hemos podido llegar a comprobar el funcionamiento del frenado regenerativo con el montaje físico debido a la falta de material, tanto un elemento para acoplar el motor BLDC con uno DC., como una batería a la que inyectar la energía que sobra.

También podrían añadirse más funciones a las instaladas en el montaje, como un regulador PI, que soporte las variaciones al cambiar las cargas. En la simulación lo hemos logrado incluir.

Además el tutor del TFG me ha manifestado su propósito de utilizar este proyecto como contenido didáctico en sus clases, para que sus alumnos



vislumbren con otra perspectiva una parte del todo el conjunto que conforma un VE.

Este tipo de sistema es polivalente para otro tipo de aplicaciones a parte de los vehículos de tracción eléctrica, tales como drones, juguetes, la industria o defensa.

Como hemos mencionado, al ser de montaje personal, hay infinidad de cosas que se podrían mejorar. Una de ellas sería hacer la placa en montaje PCB con un diseño más compacto. Los componentes elegidos se sustituirían por otros de mayores capacidades, y de esta manera poder mover motores de mayor potencia. Por último referente al software podría crearse un código más depurado utilizando los puertos serie de Arduino.

Con este TFG pretendo aportar una pequeña contribución para que se siga investigando y puedan prosperar en un futuro no muy lejano este tipo de automóviles, no sólo por las mejoras tecnológicas que presentan; sino para ayudar a preservar el medio ambiente. Este es un objetivo del que debemos ser conscientes en el presente, y reflexionar que constituye uno de los mejores legados que podemos dejar a las generaciones venideras.



FUENTES

BIBLIOGRAFÍA

El vehículo eléctrico: tecnología, desarrollo y perspectiva: Jose Fullea García, Francisco Trinidad López, J.Carlos Amasorrain Zavala, Mikel Sanzberro Iriarte.

Build Your Own Electric Motorcycle: Carl Vogel.

Apuntes “electrónica industrial”: L. Carlos Herrero, J. Antonio Vázquez, J.Miguel Ruiz.

Arduino. Curso práctico de formación. Oscar Torrente Artero.

Arduino Internals. Dale wheat.

Revista GreenCar, edición primer trimestre 2015

PÁGINA WEB/ENLACES

<http://www.endesavehiculoelectrico.com>

<http://www.arduino.cc>

<http://arduinothedocumentary.org>

<http://www.fsf.org>

<http://www.microchip.com/>

<http://www.economist.com/>

<http://www.movele.es/>

<http://www.mcgraw-hill.es/>



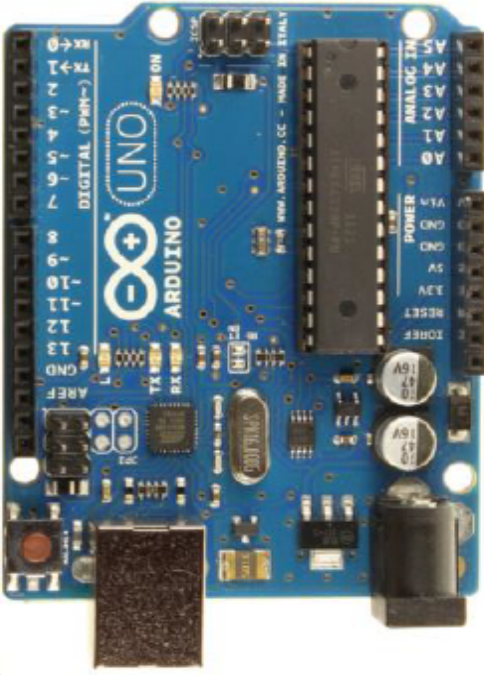
[PÁGINA INTENCIONADAMENTE EN BLANCO]



ANEXOS

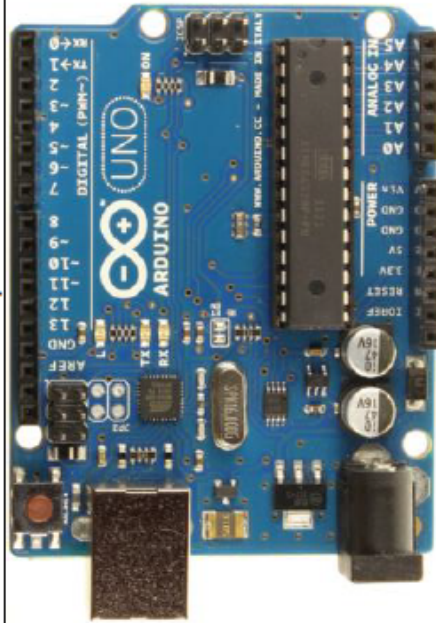
DATASHEET ARDUINO

AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

<p>ARDUINO UNO Revision 3 BOARD</p>  <p>The Arduino Uno is one of the most common and widely used Arduino processor boards. There are a wide variety of shields (plug in boards adding functionality). It is relatively inexpensive (about \$25 - \$35). The latest version as of this writing (3/2014) is Revision 3 (r3):</p> <ul style="list-style-type: none"> • Revision 2 added a pull-down resistor to the 8U2 HWB line, making it easier to put into DFU (Device Firmware Update) mode • Revision 3 added <ul style="list-style-type: none"> ◦ SDA and SCL pins are now brought out to the header near the AREF pin (upper left on picture). SDA and SCL are for the I2C interface ◦ IOREF pin (middle lower on picture that allows shields to adapt to the voltage provided ◦ Another pin not connected reserved for future use <p>The board can be powered from the USB connector (usually up to 500ma for all electronics including shield), or from the 2.1mm barrel jack using a separate power supply when you cannot connect the board to the PC's USB port.</p>	<p>Links:</p> <ul style="list-style-type: none"> • Arduino web site: http://www.arduino.cc/ • Arduino Uno overview and image source: http://arduino.cc/en/Main/arduinoBoardUno#UxNpBk2YZuG • DFU Mode (Device Firmware update) explanation: http://arduino.cc/en/Hacking/DFUProgramming8U2#UxNqXE2YZuE • Arduino Uno schematic: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf • Arduino Uno Eagle PCB Files: http://arduino.cc/en/uploads/Main/arduino_Uno_Rev3-02-TH.zip • Eagle PCB design software (use License = "Run as Freeware"): https://www.cadsoftusa.com/download-eagle/ • Hardware Index – past and present boards: http://arduino.cc/en/Main/Boards#UxNq9U2YZuE • Specifications comparison chart: http://arduino.cc/en/Products.Compare#UxOJGk2YZuF • Board comparison chart: http://arduino.cc/en/Products.Compare#UxN6oE2YZuE • Sources <ul style="list-style-type: none"> ◦ MP3Car: http://store.mp3car.com/SearchResults.asp?Search=arduino ◦ Sparkfun: https://www.sparkfun.com/ ◦ Adafruit: http://www.adafruit.com/category/17 ◦ Amazon: http://www.amazon.com/s/ref=nb_sb_noss_1?url=search-alias%3Daps&field-keywords=Arduino ◦ Pololu: http://www.pololu.com/search?query=Arduino
---	--

AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 Specifications



- Microcontroller: ATmega328
- Operating Voltage: 5V
- Uno Board Recommended Input Voltage: 7 – 12 V
- Uno Board Input Voltage Limits: 6 – 20 V
- Digital I/O Pins: 14 total – 6 of which can be PWM
- Analog Input Pins: 6
- Maximum DC Current per I/O pin at 5VDC: 40ma
- Maximum DC Current per I/O pin at 3.3 VDC: 50ma
- Flash Memory: 32KB (0.5KB used by bootloader)
- SRAM Memory: 2KB
- EEPROM: 1KB
- Clock Speed: 16 MHz

Links:

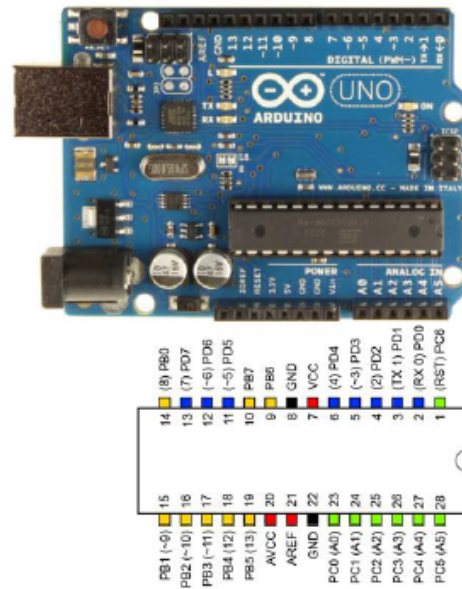
- Arduino specifications and image page: <http://arduino.cc/en/Main/arduinoBoardUno#UxOOLk2YZuH>

AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 Processor Peripherals (Atmel ATmega 328)	 <ul style="list-style-type: none"> • Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode • One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode • Real Time Counter with Separate Oscillator • Six PWM channels • Six channel 10 bit ADC including temperature measurement • Programmable Serial USART • Master/Slave SPI Serial Interface • Byte-oriented 2 wire Serial Interface (Philips I2C compatible) • Programmable Watchdog Timer with Separate On-chip Oscillator • On-chip Analog Comparator
Links: <ul style="list-style-type: none"> • Source of above diagram: http://tekkpinoy.com/wp-content/uploads/2013/10/1.jpg • AT Mega 328 datasheet: http://www.atmel.com/Images/doc8161.pdf 	

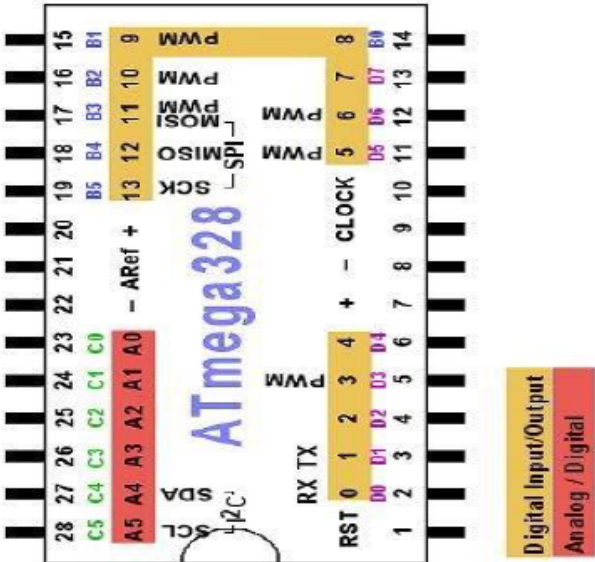
AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 and ATmega328 processor





AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 Processor Pinout (Atmel ATmega 328) – Commonly Used	
	<p>Pin Definition</p> <ul style="list-style-type: none">• PORT B (PB0 – PB7) is an 8 bit bidirectional I/O port with internal pull-ups. Processor pins 14 – 17 bring PB0 to PB5 out<ul style="list-style-type: none">◦ PB0 – PB5 are also interrupts 0-5 respectively◦ PB1 can also be used as a PWM output◦ PB2 can also be SPI Bus Master Slave Select (*SS) or PWM output◦ PB3 can also be or SPI Bus Master Out/Slave In (MOSI) or PWM output◦ PB4 can also be SPI Bus Master In/Slave Out (MISO)◦ PB5 can also be SPI Bus Master Clock Input (SCK)◦ PB6 and PB7 are brought out on Processor pins 9 and 10 for the crystal clock oscillator• PORT C (PC0 – PC5) is a 7 bit bidirectional I/O port with internal pull-up resistors. Processor pins 23 – 28 bring PC0 to PC5 out.<ul style="list-style-type: none">◦ PC0 – PC5 are also interrupts 8-13 respectively◦ PC0 – PC5 can also be used as A/D inputs◦ PC4 and PC5 can also be used as SDA and SCL for I2C◦ PC6 is brought out on processor pin 1 as reset• PORT D (D0 – D7) is an 8 bit bidirectional I/O port with internal pull-ups. Processor pins 2 – 6 and 11 – 13 bring all pins out<ul style="list-style-type: none">◦ PD0 can also be USART Input (RXD)◦ PD1 can also be USART Output (TXD)◦ PD3 can also be used as a PWM output◦ PD5 can also be used as a PWM output◦ PD6 can also be used as a PWM output
<p>Links:</p> <ul style="list-style-type: none">• Source of above diagram: http://www.hobbytronics.co.uk/arduino-atmega328-pinout• AT Mega 328 datasheet: http://www.atmel.com/Images/doc8161.pdf	

AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 Pinout (Uno PCB) – Commonly Used Features are printed on Silkscreen	<p>The Arduino Uno pinout is printed in the silkscreen on the top of the part. While this pinout is a good start, it does not explain the complete story – but it does give a good beginning. At first you use mainly the pins in the female headers at the edge of the board (top and bottom in the photo), plus USB and maybe power</p> <ul style="list-style-type: none"> • Tx and Rx are serial UART pins used for RS-232 and USB communications • I2C is another serial communications method using a bidirectional data line (SDA) and a clock line (SCL) • SPI is another serial communications method using one line for the master to transmit (MOSI – Master Out Slave In), another for the master to receive (MISO), and a third as a clock (SCK) • A/D in Analogue to Digital this input converts an analogue voltage in to a digital representation • PWM (Pulse Width Modulator) is used to create a square wave with a specific duty cycle (high time vs low time) • ICSP is the In Circuit Serial Programming – another way to program the processor • Vcc is the voltage supplied to the processor (+5VDC regulated from the higher input voltage) • 3.3VDC is a regulated voltage (from the higher input voltage) for peripherals needing that voltage – 50ma maximum • IOREF provides a voltage reference so shields can select the proper power source • AREF is a reference INPUT voltage used by the A/Ds • GND is the ground reference • RESET resets the processor (and some peripherals)
	<p>Links:</p> <ul style="list-style-type: none"> • Source of above diagram: http://www.adafruit.com/blog/2012/05/25/handy-arduino-r3-pinout-diagram/ • Description of pin usage: http://www.gammon.com.au/forum/?id=11473 • Arduino Uno Pin Mapping: http://arduino.cc/en/Hacking/PinMapping168#_UxOjlk2YZuE • Description of Arduino Serial: http://arduino.cc/en/reference/serial#_UxOMKk2YZuE • Description of the Arduino SPI functions and library: http://arduino.cc/en/Reference/SPI#_UxOPLk2YZuE • Description of Arduino A/D: http://arduino.cc/en/Tutorial/AnalogInputPins#_UxOM7k2YZuE • Description of Arduino PWM: http://arduino.cc/en/Tutorial/PWM#_UxOLz02YZuE • Tutorial on ISP: http://arduino.cc/en/Tutorial/ArduinoISP#_UxOUSk2YZuE • Tutorial on the AREF pin: http://tronixstuff.com/2013/12/12/arduino-tutorials-chapter-22-aref-pin/



AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

ARDUINO UNO Revision 3 Processor Pinout (Atmel ATmega 328) – Other functions	
<div><div>Arduino function</div><div>reset (PCINT4/RESET) PD6</div><div>digital pin 0 (RX) (PCINT16/XTAL1/TOSC1) PB6</div><div>digital pin 1 (TX) (PCINT17/XTAL2/TOSC2) PB7</div><div>digital pin 2 (PCINT18/INT0) PD2</div><div>digital pin 3 (PWM) (PCINT19/OC2B/INT1) PD3</div><div>digital pin 4 (PWM) (PCINT20/OC2A/INT0) PD4</div><div>VCC</div><div>GND</div><div>crystal</div><div>crystal</div><div>digital pin 5 (PWM) (PCINT21/OC0B/INT1) PD5</div><div>digital pin 6 (PWM) (PCINT22/OC0A/INT0) PD6</div><div>digital pin 7 (PCINT23/INT1) PD7</div><div>digital pin 8 (PCINT24/INT0) PB0</div></div>	<div><div>Arduino function</div><div>analog input 5</div><div>analog input 4</div><div>analog input 3</div><div>analog input 2</div><div>analog input 1</div><div>analog input 0</div><div>GND</div><div>analog reference</div><div>VCC</div><div>digital pin 13</div><div>digital pin 12</div><div>digital pin 11 (PWM) (PCINT25/OC0A/INT0) PD5</div><div>digital pin 10 (PWM) (PCINT26/OC0B/INT1) PD6</div><div>digital pin 9 (PWM) (PCINT27/OC0A/INT0) PD7</div></div> <div><div>Pin Definition</div><div><ul style="list-style-type: none">PORT B pins, in addition to digital I/O have other uses<ul style="list-style-type: none">PB0 can also be the divided system clock output (CLKO) or Timer/Counter 1 Input Capture (ICP1)PB1 can also be Timer/Counter1 Output Compare Match A (OC1A) outPB2 can also be Timer/Counter1 Output Compare Match B (OC1B)PB3 can also be Timer/Counter2 Output Compare Match A out(OC2A)Port D pins, in addition to digital I/O have other uses<ul style="list-style-type: none">PD3 is also Timer/Counter2 Output Compare Match B Output (OC2B)PD4 is also Timer/Counter0 External Counter Input (T0) or USART External Clock Input/Output (XCK)PD5 is also Timer/Counter0 Output Compare Match B Output (OC0B) and Timer/Counter 1 External Counter InputPD6 can also be Analog Comparator Positive In (AIN0)PD7 can also be Analog Comparator Negative In (AIN1)</div></div>
<div><div>Links:</div><ul style="list-style-type: none">Source of above diagram: http://nearbus.net/wiki/index.php?title=Atmega_328_PinoutAT Mega 328 datasheet: http://www.atmel.com/images/doc8161.pdf<div>NOTE: A single diagram showing all features of the Arduino Uno and the Atmel ATmega328 processor is shown in Appendix A</div></div>	



AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

APPENDIX A

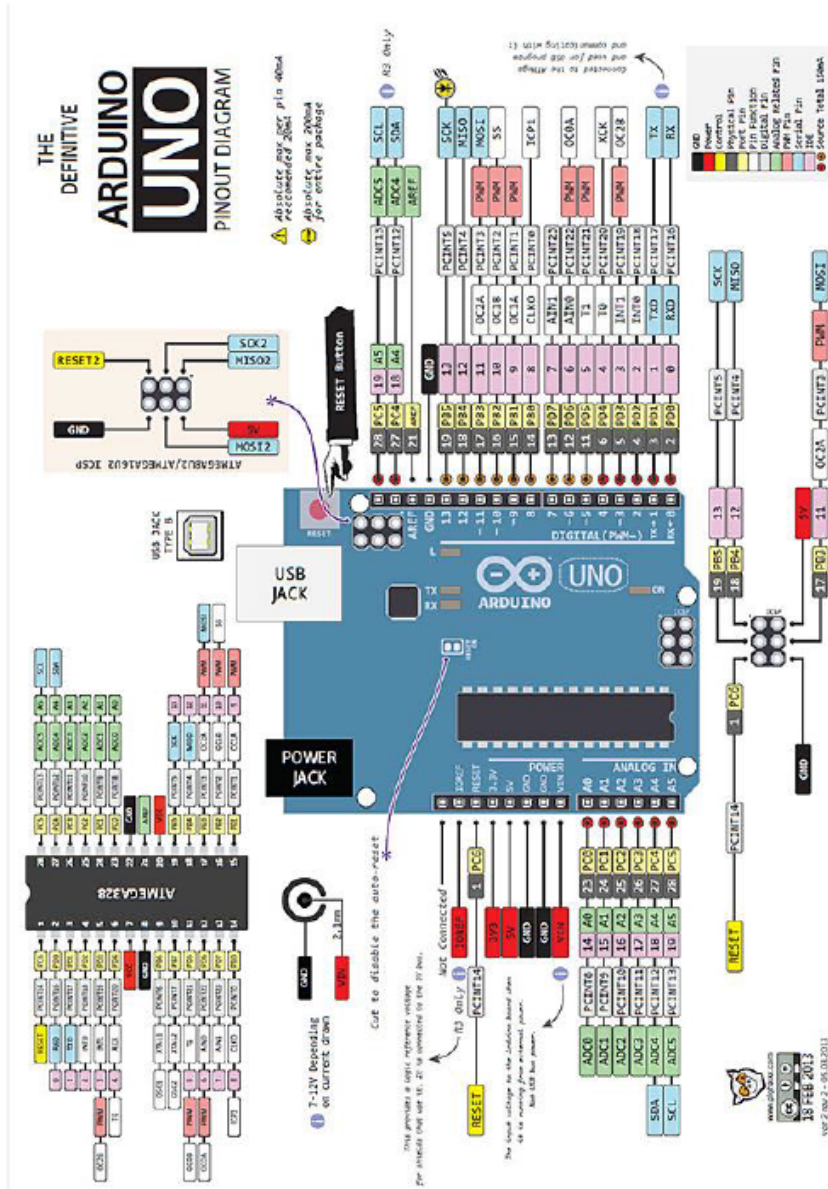


Diagram from: <http://arduino-info.wikispaces.com/file/view/ArduinoUNO-900.jpg/421496636/ArduinoUNO-900.jpg>



AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

APPENDIX B

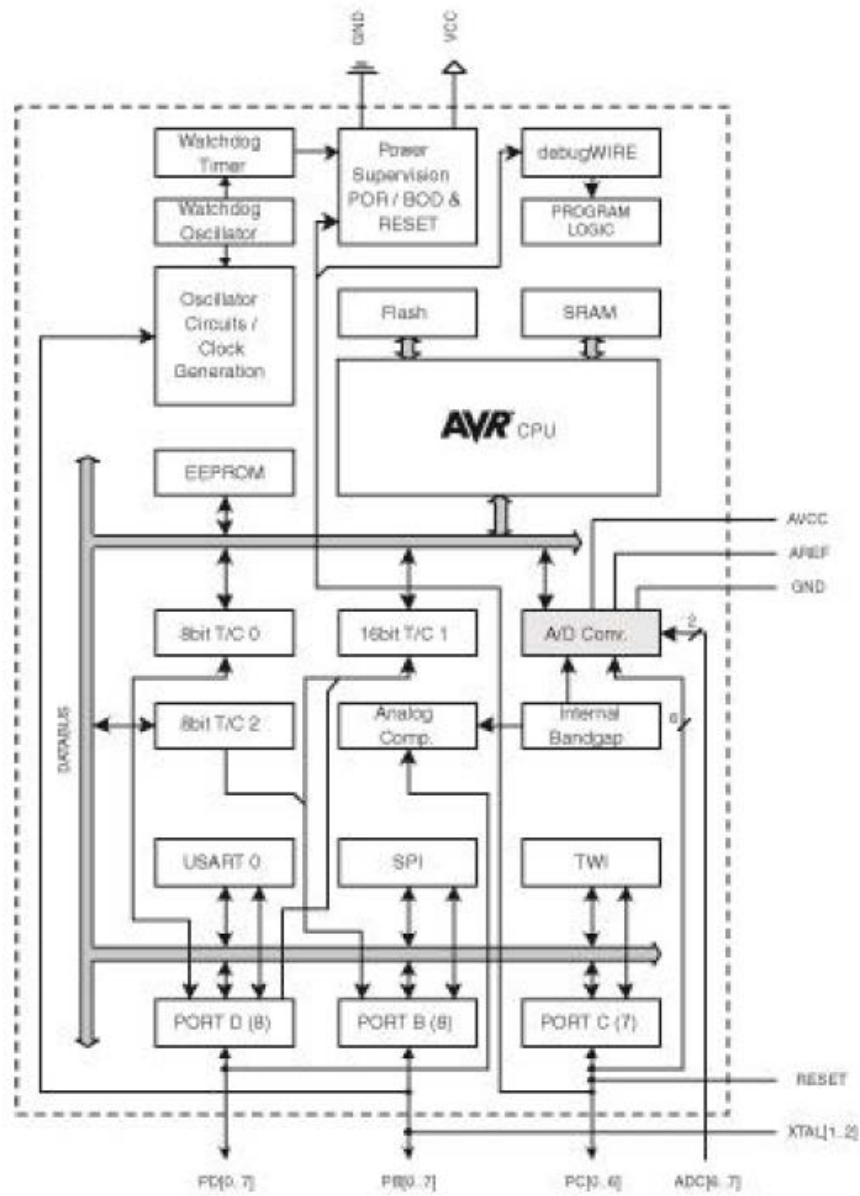
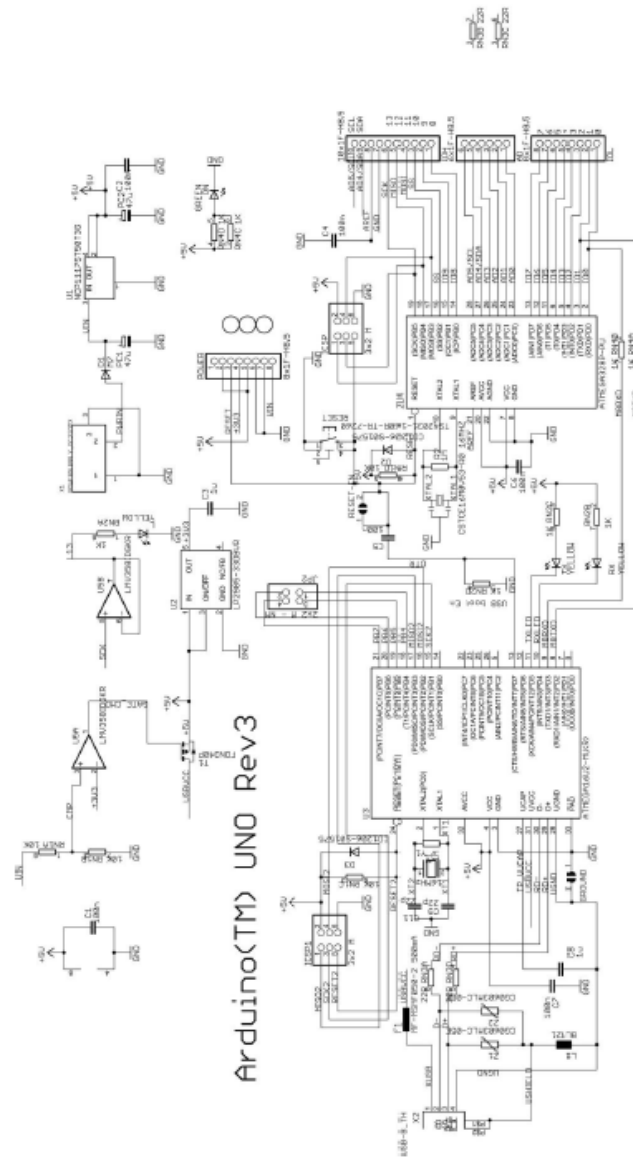


Diagram from document at: <http://www.atmel.com/Images/doc8161.pdf>



AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

APPENDIX C



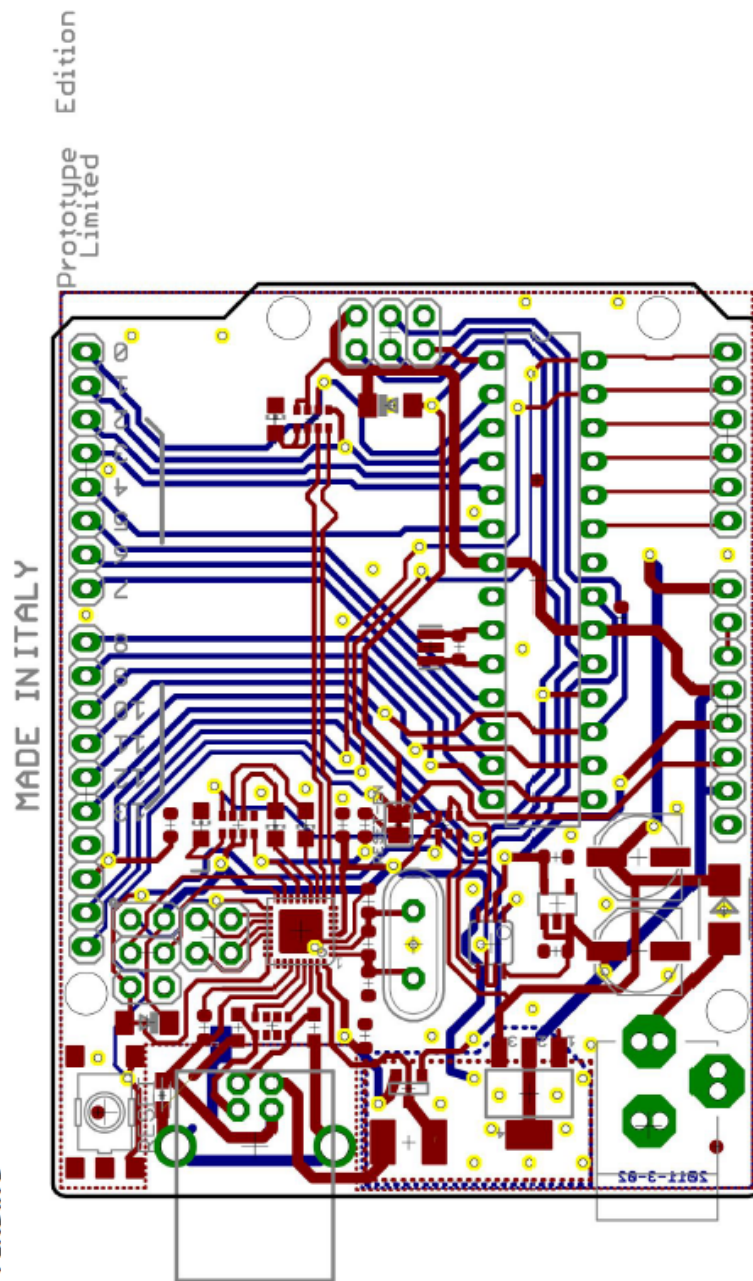
Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. ARDUINO DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ARDUINO MAY MAKE CHANGES TO SPECIFICATIONS AND PRODUCT DESCRIPTIONS AT ANY TIME, WITHOUT NOTICE. THE CUSTOMER MUST NOT RELY ON THE ABSENCE OR CHARACTERISTICS OF ANY FEATURES OR INSTRUCTIONS MARKED "RESERVED" OR "UNDEFINED." ARDUINO RESERVES THESE FOR FUTURE DEFINITION AND SHALL HAVE NO RESPONSIBILITY WHATSOEVER FOR CONFLICTS OR INCOMPATIBILITIES ARISING FROM FUTURE CHANGES TO THEM. THE PRODUCT INFORMATION ON THE WEB SITE OR MATERIALS IS SUBJECT TO CHANGE WITHOUT NOTICE. DO NOT FINALIZE A DESIGN WITH THIS INFORMATION. ARDUINO IS A REGISTERED TRADEMARK.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Diagram from document at: http://arduino.cc/en/uploads/Main/Arduino_Uno_Rev3-schematic.pdf

AIAA OC Rocketry (Revision 3 April 27, 2014 - <http://aiaaocrocketry.org>)

APPENDIX D



From arduino_Uno_Rev3-02-TH.zip file at <http://arduino.cc/en/Main/ArduinoBoardUno#Uxk9qk2YYpA>

Eagle PCB software: Eagle PCB design software (use License = "Run as Freeware"): <https://www.cadsoftusa.com/download-eagle/>

DATASHEET INVERSOR

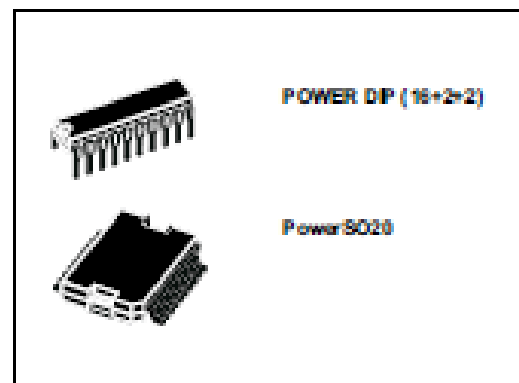


L6234

Three phase motor driver

Features

- Supply voltage from 7 to 52 V
- 5 A peak current
- $R_{DS(on)}$ 0.3 Ω typ. value at 25 °C
- Cross conduction protection
- TTL compatible driver
- Operating frequency up to 150 kHz
- Thermal shutdown
- Intrinsic fast free wheeling diodes
- Input and enable function for each half bridge
- 10 V external reference available



Description

The L6234 is a triple half bridge to drive a brushless DC motor.

It is realized in BCDmultipower technology which combines isolated DMOS power transistors with CMOS and Bipolar circuits on the same chip.

By using mixed technology it has been possible to optimize the logic circuitry and the power stage to achieve the best possible performance.

The output DMOS transistors can sustain a very high current due to the fact that the DMOS structure is not affected by the second breakdown effect, the RMS maximum current is practically limited by the dissipation capability of the package.

All the logic inputs are TTL, CMOS and μP compatible. Each channel is controlled by two separate logic input.

L6234 is available in 20 pin PowerDIP package (16+2+2) and in PowerSO20.

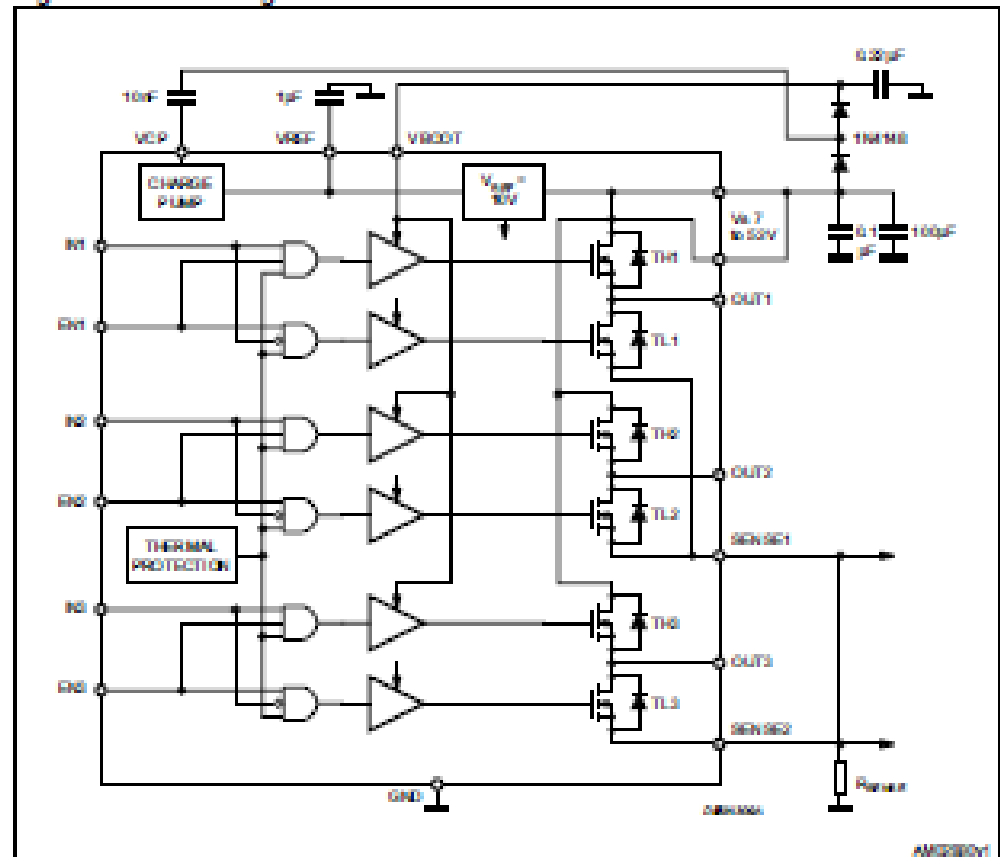
Table 1. Device summary

Order code	Package	Packing
L6234	PowerDIP20	Tube
L6234PD	PowerSO20	Tube
L6234PD013TR	PowerSO20	Tape and reel



Contents

1	Block diagram	3
2	Pin connections	4
3	Thermal data	5
4	Maximum ratings	6
4.1	Recommended operating conditions	6
5	Electrical characteristics	7
6	Circuit description	8
7	Typical characteristics	9
8	Mechanical data	12
9	Revision history	15





2 Pin connections

Figure 2. Pin connections

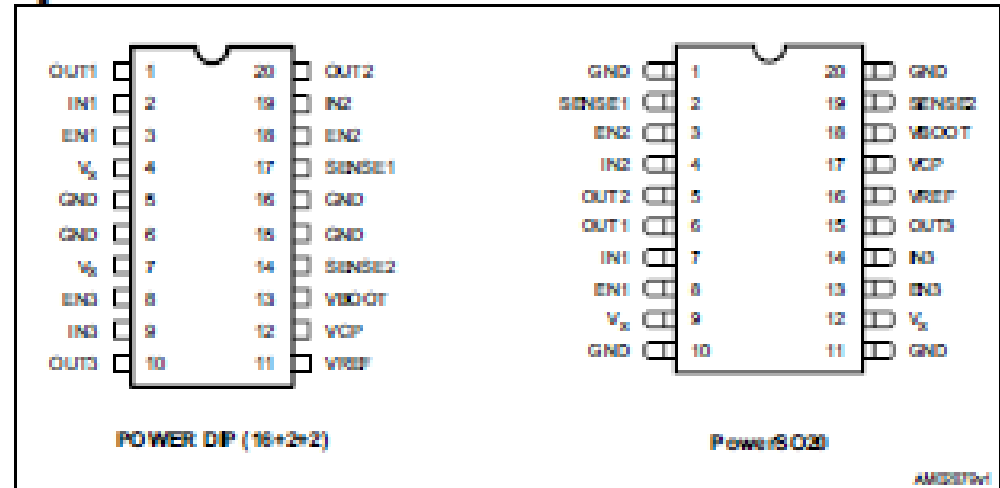


Table 2. Pin functions

PowerDIP	PowerSO20	Name	Function
1 20 10	6 5 15	OUT 1 OUT 2 OUT 3	Output of the channels 1/2/3.
2 19 9	7 4 14	IN 1 IN 2 IN 3	Logic input of channels 1/2/3. A logic HIGH level (when the corresponding EN pin is HIGH) switches ON the upper DMOS Power Transistor, while a logic LOW switches ON the corresponding low side DMOS Power.
3 18 8	8 3 13	EN 1 EN 2 EN 3	Enable of the channels 1/2/3. A logic LOW level on this pin switches off both power DMOS of the related channel.
4,7	9, 12	V _s	Power supply voltage.
14	19	SENSE2	A sense resistor connected to this pin provides feedback for motor current control for the bridge 3.
17	2	SENSE1	A sense resistor connected to this pin provides feedback for motor current control for the bridges 1 and 2.
11	16	VREF	Internal voltage reference. A capacitor connected from this pin to GND increases the stability of the Power DMOS drive circuit.
12	17	VCP	Bootstrap oscillator. Oscillator output for the external charge pump.
13	18	VBOOT	Overvoltage input to drive the upper DMOS
5,6 15,16	1,10 11,20	GND	Common ground terminal. In PowerDIP and SO packages these pins are used to dissipate the heat forward the PCB.



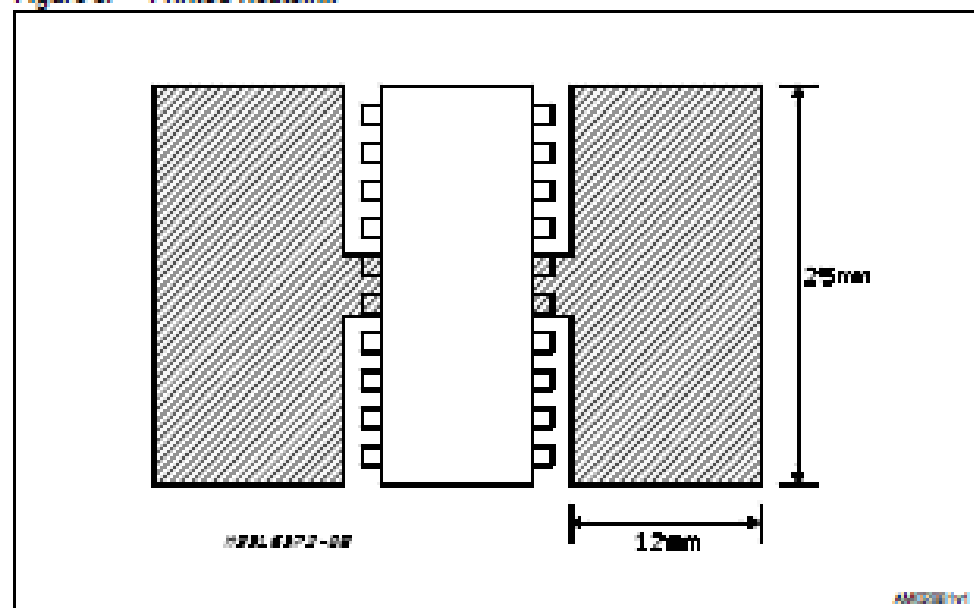
3 Thermal data

Table 3. Thermal data

Symbol	Parameter	DIP16+2+2	PowerSO20	Unit
$R_{th(j-pin)}$	Thermal resistance, junction to pin	12 ⁽¹⁾	—	°C/W
$R_{th(j-amb1)}$	Thermal resistance, junction to ambient	40 ⁽²⁾	—	°C/W
$R_{th(j-amb2)}$	Thermal resistance, (junction characteristics) to ambient	50 ⁽²⁾	—	°C/W
$R_{th(j-case)}$	Thermal resistance junction-case	—	1.5	°C/W

1. The thermal resistance is related to the thermal path from the dissipating region on the top surface of the silicon chip, to the points along the four central pins of the package, at a distance of 1.5 mm away from the stand-offs.
2. If a dissipating surface, thick at least 35 mm, and with a surface similar or bigger than the one shown in [Figure 3](#), is created making use of the printed circuit. Such heatsinking surface is considered on the bottom side of an horizontal PCB [worst case].
3. If the power dissipating pins (the four central ones), as well as the others, have a minimum thermal connection with the external world (very thin stripes only) so that the dissipation takes place through still air and through the PCB itself. It is the same situation of note 2, without any heatsinking surface created on purpose on the board.

Figure 3. Printed Heatsink





4 Maximum ratings

Table 4. Absolute maximum ratings

Symbol	Parameter	Value	Unit
V_{IS}	Power supply voltage	52	V
V_{IN}, V_{EN}	Input enable voltage	- 0.3 to 7	V
I_{peak}	Pulsed output current ⁽¹⁾	5	A
V_{SENSE}	Sensing voltage (DC voltage)	-1 to 4	V
V_{boot}	Bootstrap peak voltage	62	V
V_{OD}	Differential output voltage (between any of the 3 OUT pins)	60	V
f_C	Commutation frequency	150	kHz
V_{REF}	Reference voltage	12	V
P_{tot}	Total power dissipation L6234PD, $T_A = 70^\circ\text{C}$	2.3	W
P_{tot}	Total power dissipation L6234, $T_A = 70^\circ\text{C}$	1.6 ⁽²⁾	W
T_{stg}, T_J	Storage and junction temperature range	-40 to 150	$^\circ\text{C}$

1. Pulse width limited only by junction temperature and the transient thermal impedance

2. Mounted on board with minimized copper area

4.1 Recommended operating conditions

Table 5. Recommended operating conditions

Symbol	Parameter	Value	Unit
V_{IS}	Supply voltage	7 to 42	V
V_{OD}	Peak to peak differential voltage (between any of the 3 out pins)	52	V
I_{out}	DC output current powerSO20 ($T_A = 25^\circ\text{C}$)	4	A
	DC output current powerDIP ($T_A = 25^\circ\text{C}$) with infinite heatsink	2.8	A
V_{SENSE}	Sensing voltage (pulsed $t_{pw} < 300$ nsec)	-4 to 4	V
	Sensing voltage (DC)	-1 to 1	V
T_J	Junction temperature range	-40 to 125	$^\circ\text{C}$



5 Electrical characteristics

$V_S = 42\text{ V}$; $T_J = 25\text{ }^{\circ}\text{C}$ unless otherwise specified.

Table 6. Electrical characteristics

Symbol	Parameter	Test condition	Min.	Typ.	Max.	Unit
V_S	Supply voltage		7		52	V
V_{ref}	Reference voltage			10		V
I_S	Quiescent supply current			6.5		mA
T_S	Thermal shutdown		150			$^{\circ}\text{C}$
T_D	Dead time protection			300		ns
Output dmos transistor						
I_{DSS}	Leakage current				1	mA
$R_{DS(on)}$	ON resistance			0.3		Ω
Source drain diode						
V_{SD}	Forward ON voltage	$I_{SD} = 4\text{ A}$; $EN = \text{LOW}$		1.2		V
T_{RR}	Reverse recovery time	$I_F = 4\text{ A}$		900		ns
T_{pr}	Forward recovery time			200		ns
Logic levels						
V_{INL}, V_{ENL}	Input LOW voltage		-0.3		0.8	V
V_{INH}, V_{ENH}	Input HIGH voltage		2		7	V
I_{INL}, I_{ENL}	Input LOW current	$V_{IN}, V_{EN} = \text{L}$			-10	μA
I_{INH}, I_{ENH}	Input HIGH current	$V_{IN}, V_{EN} = \text{H}$		30		μA



6 Circuit description

L6234 is a triple half bridge designed to drive brushless DC motors. Each half bridge has 2 power DMOS transistors with $R_{DS(on)} = 0.3 \Omega$.

The 3 half bridges can be controlled independently by means of the 3 inputs IN1, IN2, IN3 and the 3 inputs EN1, EN2, and EN3. An external connection to the 3 common low side DMOS sources is provided to connect a sensing resistor for constant current chopping application.

The driving stage and the logic stage are designed to work from 7 V to 52 V.

7 Typical characteristics

Figure 4. Quiescent current vs. supply voltage

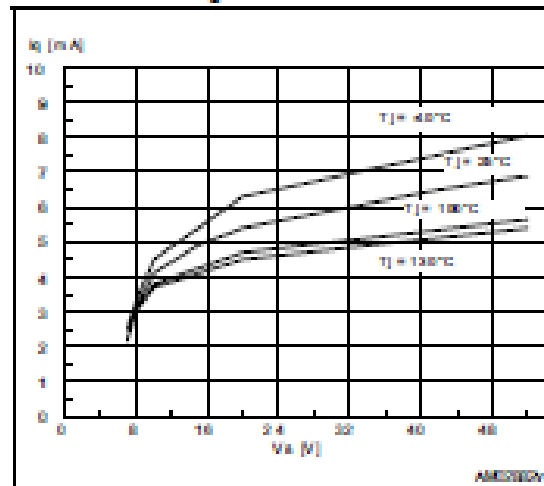


Figure 5. Normalized quiescent current vs. switching frequency

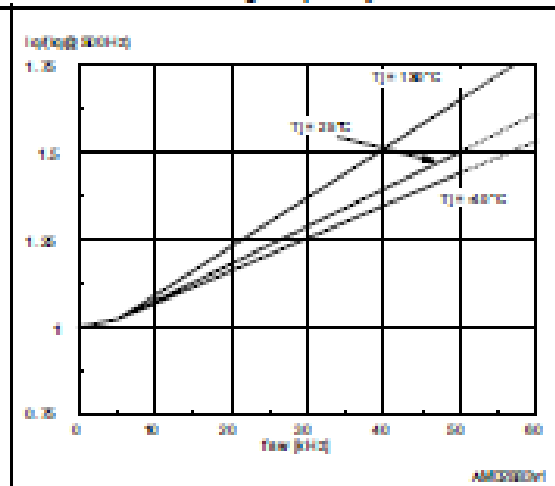


Figure 6. Typical $R_{DS(on)}$ vs. supply voltage

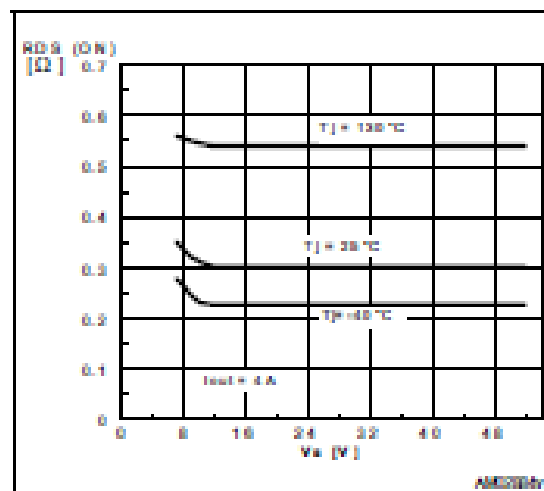
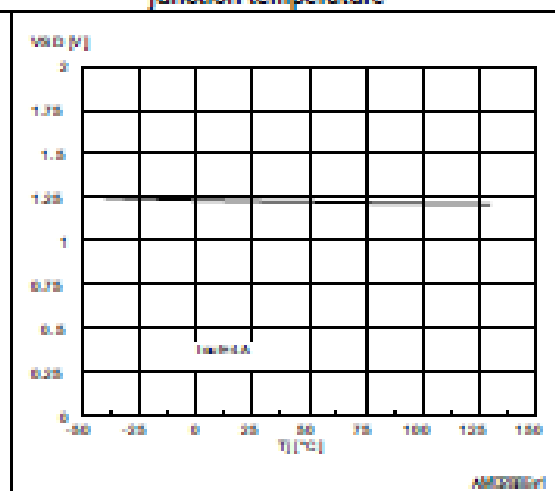


Figure 7. Source drain forward on voltage vs. junction temperature



Typical characteristics

L6234

Figure 8. Typical diode forward ON characteristics

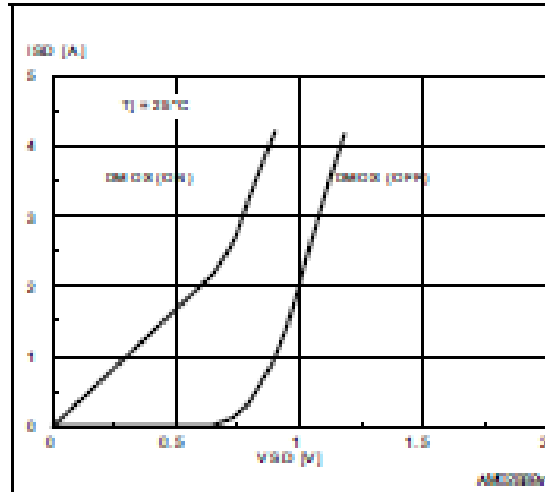


Figure 9. Reference voltage vs. supply voltage

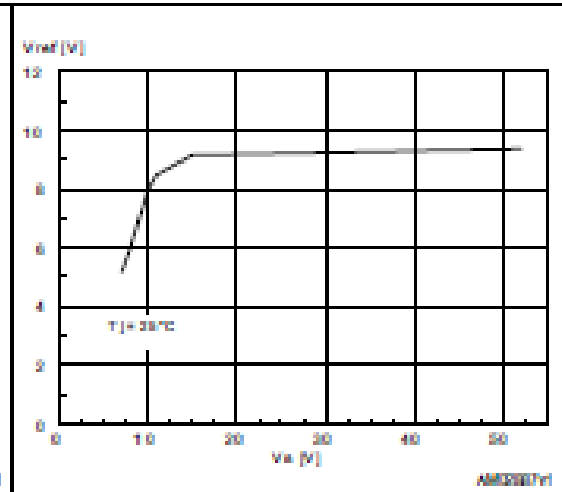


Figure 10. Reference voltage vs. junction temperature

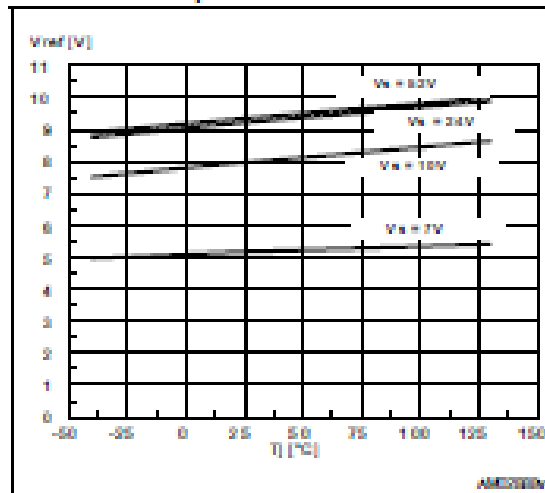
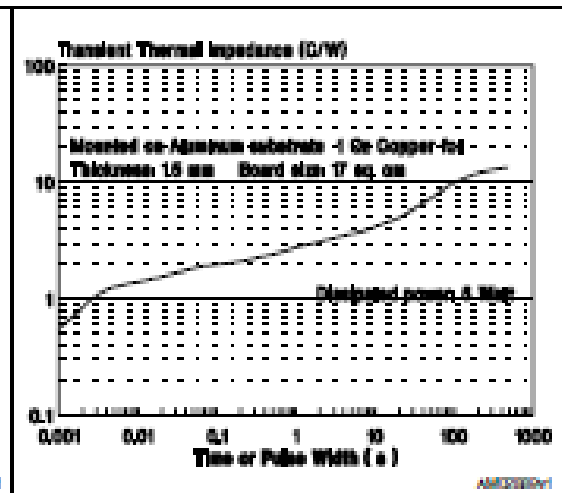


Figure 11. PowerSO-20 transient thermal resistance





8 Mechanical data

In order to meet environmental requirements, ST offers these devices in different grades of ECOPACK® packages, depending on their level of environmental compliance. ECOPACK® specifications, grade definitions and product status are available at: www.st.com. ECOPACK® is an ST trademark.

Table 7. PowersO20 mechanical data

Dim.	mm		
	Min.	Typ.	Max.
A			3.6
a1	0.1		0.3
a2			3.3
a3	0		0.1
b	0.4		0.53
c	0.23		0.32
D (1)	15.8		16
D1	9.4		9.8
E	13.9		14.5
e		1.27	
e3		11.43	
E1 (1)	10.9		11.1
E2			2.9
E3	5.8		6.2
G	0		0.1
H	15.5		15.9
h			1.1
L	0.8		1.1
N	8° (typ.)		
S	8° (max.)		
T		10	



Technical drawing of a 12-pin D-sub connector. The drawing includes three main views: a front view (top left), a top view (bottom left), and a bottom view (bottom right). Dimensions are provided in millimeters (mm) and inches (in).

Front View (Top Left): Shows the connector's profile with dimensions: H (height), A (height of the shell), A_2 (height of the mounting flange), B (width of the mounting flange), E (length of the shell), F (length of the mounting flange), and D (total length). A callout for "DETAIL A" points to the shell profile.

Top View (Bottom Left): Shows the connector's footprint with dimensions: D (total length), W (width), 20 (width of the mounting flange), 10 (width of the shell), E_1 (height of the mounting flange), T (thickness of the mounting flange), and $h \times 45$ (height of the mounting flange).

Bottom View (Bottom Right): Shows the connector's footprint with dimensions: D_1 (total length), E_3 (height of the mounting flange), and D (total length).

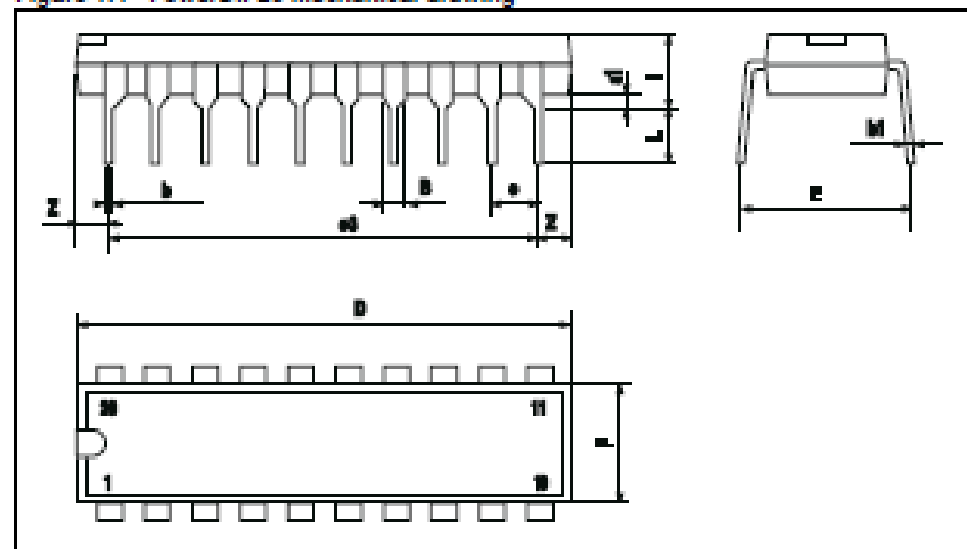
Details:

- DETAIL A:** Shows the shell profile with dimensions 1.60 and 1.60 .
- DETAIL B:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL C:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL D:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL E:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL F:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL G:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL H:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL I:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL J:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL K:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL L:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL M:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL N:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL O:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL P:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL Q:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL R:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL S:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL T:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL U:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL V:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL W:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL X:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL Y:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .
- DETAIL Z:** Shows the mounting flange profile with dimensions 1.60 and 1.60 .

Table 8. PowerDIP20 mechanical data

Dim.	mm		
	Min.	Typ.	Max.
a1	0.51		
B	0.85		1.40
b		0.50	
b1	0.38		0.50
D			24.80
E		8.80	
e		2.54	
e3		22.86	
F			7.10
I			5.10
L		3.30	
Z			1.27

Figure 17. PowerDIP20 mechanical drawing





9 Revision history

Table 9. Document revision history

Date	Revision	Changes
01-Aug-2003	9	
15-Nov-2011	10	Updated Features in coverpage and Table 4



L6234

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST's terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST'S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER'S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2011 STMicroelectronics - All rights reserved


STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com







Sample Motor Data Sheet

Date: 2/14/02

Customer	Microchip	Model Number	DMB0224C10002
		Serial #	12482

L-L Resistance (R_{LL}) Ohms :	4.03	Electrical Time Constant (τ_e) mSec. :	1.14
L-L Inductance (L_{LL}) mH at 1KHz :	4.60	Mechanical Time Constant (τ_m) mSec. :	3.74
Torque Constant (K_t) oz.in./Amp :	9.79	Thermal Resistance ($R_{\theta JA}$) °C/Watt :	4.78
Voltage Constant (K_v) V/amp/KHz :	7.24	Thermal Time Constant (θ_{JA}) min. :	16
Amb. Temp. (°C) :	22.7	Rotor Inertia (J_r) oz-in-sec ² :	0.000628
		Stack Length:	0.75

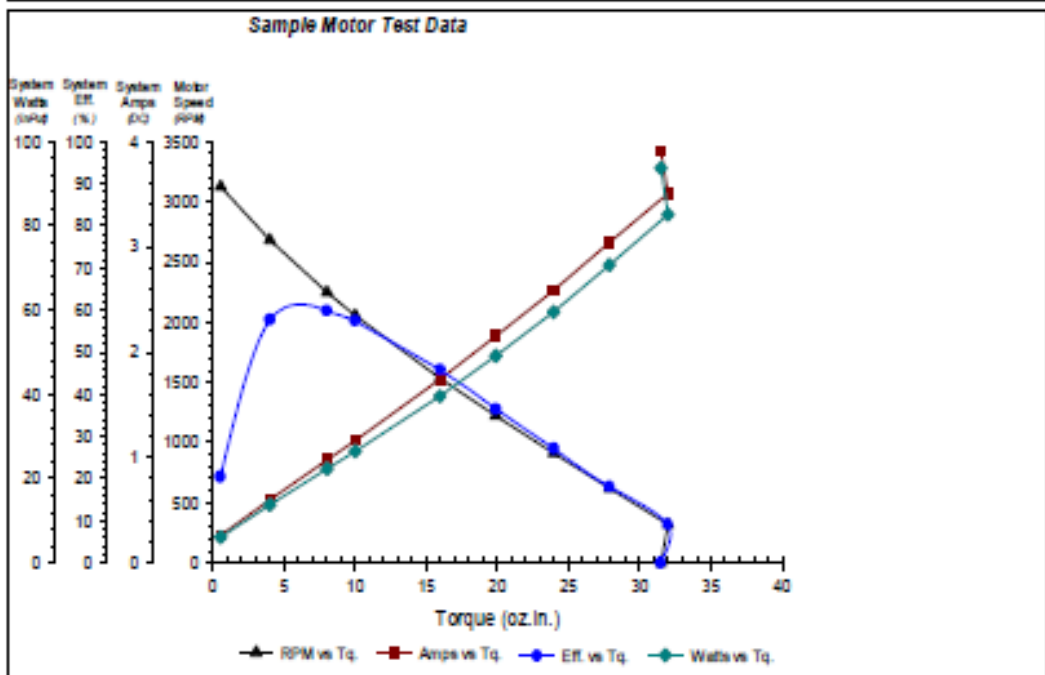
Notes:

Speed / Torque Test Data -Control Input set at 100% duty cycle.

Load	Volts (DC)	Amps (DC)	Watts (DC)	Speed (RPM)	Torque (oz.in.)	Output (watts)	Output (HP)	Eff. (%)	
1	24.01	0.26	6.11	3125	0.54	1.25	0.002	20.4	
2	24.01	0.60	13.76	2683	4.00	7.94	0.011	57.7	
3	24.01	0.98	22.29	2248	8.02	13.34	0.018	59.9	
4	24.01	1.16	26.46	2054	10.00	15.20	0.020	57.4	Max Continuous Rating
5	24.02	1.74	39.56	1534	15.96	18.12	0.024	45.8	
6	24.02	2.15	49.16	1222	19.88	17.98	0.024	36.6	
7	24.03	2.59	59.55	913	23.94	16.18	0.022	27.2	
8	24.03	3.04	70.70	621	27.86	12.80	0.017	18.1	
9	24.04	3.51	82.69	319	31.98	7.55	0.010	9.1	
10	24.04	3.91	93.72	0	31.48	0.00	0.000	0.0	

Special Load Points

1									
2									



This motor is intended for sampling and customer approval only. No application fitness approval is implied, as that can only be determined by the customer. These data represent performance of a single sample motor. These values are not to be construed as guaranteed values.

Hurst Mfg.
Company Confidential

Motor Test Data

2/18/03



EXTERNAL CONTROL MODULE DATA SHEET

Description: The External Control Module simplifies the connection of an external motor drive to the Dynamo motor by providing the user with a standard set of hall signals, numerous encoder options, and a high current connector for the motor phase windings. The module is compatible with external motor drives using a 10 to 48Vdc power supply. The External Control Module provides a standard system for rotor position sensing required by many brushless motor drives. Three hall sensors spaced 120 electrical degrees apart, sense a magnetic disk, which is synchronized to the rotor of the motor. The hall signals can be used to provide inexpensive speed feedback to the motor drive, or for more precise control a wide array of integral two channel quadrature encoder options are available. The quadrature nature of an encoder allows the user to determine the direction of motor rotation as well as speed.

Environment: The NT Dynamo uses a TENV (totally enclosed non-ventilated) non-gasket construction. Installation and operating conditions should not exceed the recommended values for humidity and temperature. Contact the Hurst engineering department regarding any special installation issues you may have regarding vapors, oils or dust.

Storage Temp.: 32-158°F (0-70°C) **Humidity:** 90% Max. Non-condensing **Operating Temp.:** 32-104°F (0-40°C)

Power: Power to the motor windings is via the four pin connector. A regulated DC supply must be provided for the encoder and hall devices. Observe the correct polarity when making these connections. For maximum flexibility and noise immunity, the hall and encoder power supplies are separated. Excessive amounts of voltage ripple can cause shortened product life.

Motor Windings:

Minimum DC Voltage: 10Vdc

Maximum DC Voltage: 48Vdc

Encoder:

Minimum DC Voltage: 4.75Vdc

Maximum DC Voltage: 5.25Vdc

Halls:

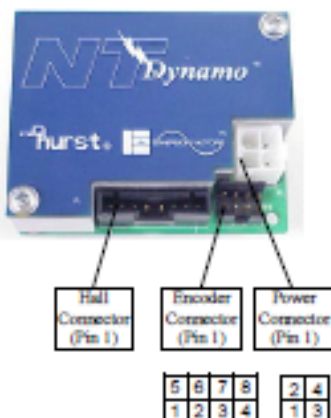
Minimum DC Voltage: 4.2Vdc

Maximum DC Voltage: 24Vdc

Connector	Pin #	Function	Mating Connector	Mating Terminal	Recommended Wire Size	Cable Length
Power	1	Phase C	Molex 39-01-2040	Molex 39-00-0039	22 AWG	30 ft. Max ¹
	2	Phase B				
	3	Phase A				
	4	Gnd				
Hall	1	V _{cc}	Molex 50-57-9408	Molex 16-02-0103	22 AWG	30 ft. Max ¹
	2	V _{cc} (GND)				
	3	Hall S2				
	4	Hall S1				
	5	Hall S3				
	6	N/A				
	7	N/A				
	8	N/A				
Encoder	1	+5V _{cc}	FCI 65846-016	FCI 48236-000	22 AWG	30 ft. Max ¹
	2	Encoder A				
	3	Encoder B				
	4	Encoder I				
	5	+5V _{cc} (GND)				
	6	Encoder /A				
	7	Encoder /B				
	8	Encoder /I				

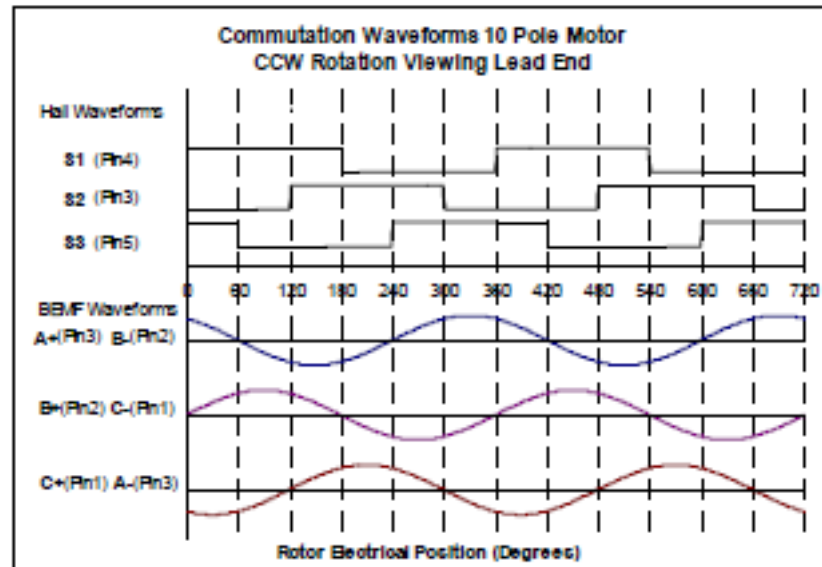
Notes

- 1) Longer cable runs may require a larger wire size to maintain the correct input voltage level and a signal amplifier / conditioner to avoid erroneous signal values. For cable runs longer than 3 ft, shielded wire is recommended.





Commutation:



Encoder:

The drive may contain an optional shaft mounted optical encoder. The encoder outputs two or four quadrature signals from which direction and speed can be determined. These outputs can be used by an external drive to close the speed loop.

For More Information Visit The Website at www.hurstmfg.com or
Contact Hurst Engineering at 812-385-2564



APÉNDICE: GLOSÁRIO DE ABREVIATURAS Y ACRÓNIMOS

AC: Alternating Current

BLDC: BrushLess Direct Current

DC: Direct Current

PCB: Printed Circuit Board

PI: Proporcional Integral

PWM: Pulse With Modulation

RF: Radio Frecuencia

SPI: Serial Peripheal Interface

VCI: Vehículo de Combustión Interna

VE: Vehículo Eléctrico

VH: Vehículo Híbrido



[PÁGINA INTENCIONADAMENTE EN BLANCO]



NOTAS